21世纪研究生教材

实用数值分析

SHIYONG SHUZHI FENXI

第二版

张光澄 张 雷 编著

SHIYONG SHUZHI FENXI



四川大学出版社

实用数值分析

第二版

SHIYONG SHUZHI FENXI


ISBN 978-7-5614-4249-4

定价：38.00 元

四川大学研究生教材建设基金重点资助项目

实用数值分析

SHIYONG SHUZHI FENXI

第二版

张光澄 张 雷 编著

四川大学出版社 

责任编辑:李川娜
责任校对:王　锋
封面设计:罗　光
责任印制:曹　琳

图书在版编目(CIP)数据

实用数值分析 / 张光澄，张雷编著. —2 版. —成都：四川大学出版社，2009.1
ISBN 978-7-5614-4249-4

Ⅰ.实… Ⅱ.①张…②张… Ⅲ.数值计算 Ⅳ.O241

中国版本图书馆 CIP 数据核字(2009)第 011605 号

内容简介

本书主要介绍计算机上常用的数值计算方法及相关的基本概念和理论.全书分为两个部分:第一部分为正文,共包含 9 章内容(含习题).第 1 章介绍算法及其基本特点和误差的基本概念;第 2 章至第 9 章介绍工程上常用的数值计算方法以及相关的基本理论.第二部分包含两个附录.附录Ⅰ主要介绍当今最流行的数学软件 Matlab 在数值计算方法、最优化方法以及数据处理等方面的应用;附录Ⅱ为习题详解和参考答案.本书突出方法,突出应用.

本书可作为高等院校理工科硕士生、工程硕士生数值分析和数值计算方法课程的教材,也可供从事相关工作的科研人员和工程技术人员参考.

书名　实用数值分析(第二版)

编　　著　张光澄　张　雷
出　　版　四川大学出版社
地　　址　成都市一环路南一段 24 号 (610065)
发　　行　四川大学出版社
书　　号　ISBN 978-7-5614-4249-4
印　　刷　四川大学印刷厂
成品尺寸　185 mm×260 mm
印　　张　20.25
字　　数　458 千字
版　　次　2009 年 2 月第 2 版
印　　次　2009 年 2 月第 1 次印刷
印　　数　0 001~1 500 册
定　　价　38.00 元

◆读者邮购本书,请与本社发行科联系。电话:85408408/85401670/85408023　邮政编码:610065
◆本社图书如有印装质量问题,请寄回出版社调换。
◆网址:www.scupress.com.cn

序

"年年岁岁花相似，年年岁岁人不同"．在世纪之交，经过又一次高教体制改革，强强合并后的新四川大学已成为我国西部地区规模最大、学科门类最齐全的新型综合性研究型大学．

作为新世纪的献礼，我校研究生教材建设基金资助的第一批研究生优秀教材正式出版了，我在此表示热烈地祝贺．

众所周知，21世纪是知识经济的世纪，国际竞争空前激烈．竞争的焦点是科学技术，竞争的核心是创新型人才，竞争的关键是国民教育．对于四川大学这样的国家重点大学而言，则要大力发展研究生教育，扩大研究生规模，注重研究生质量．

校长、教师、教材是办学中的三大要素．教材是教学改革与师生智慧的重要的物化结晶．正是基于这种思考，我校决定在以学科建设为龙头的同时，努力加强研究生的教材建设，通过各种渠道筹集了专项基金，用以资助研究生优秀教材的编写和出版．我们首次资助的是有博士学位授权点的学科专业中涉及面大、使用面宽的研究生学位平台课程的优秀教材．今后，还将陆续扩大教材基金资助的范围，包括资助我校新增加的医学门类的有关教材的出版．

这次推出的研究生教材的基本特点是：符合该学科教学大纲的基本要求，有较强的理论性和系统性．它既反映了该学科发展的新知识、新动向、新成就，也反映了我校教师在该门学科教学与科研中的成果和经验．

前人说得好：古今之成大学问、大事业者，都必须经过三种境界．"昨夜西风凋碧树，独上高楼，望尽天涯路"，此第一境也；"衣带渐宽终不悔，为伊消得人憔悴"，此第二境也；"众里寻他千百度，蓦然回首，那人却在灯火阑珊处"，此第三境也．研究生的优秀教材的建设应该算作一种"大事业"．本教材的作者们对于研究生教育改革的执着追求，令人钦佩；他们的无私奉献精神，值得赞扬；他们所取得的教学科研成果应该积极推广，使它产生应有的社会效益，为百年名校增添光彩．我希望在首批及以后陆续出版的我校研究生教材中能出现"传诸后世"的佳作，更希望我校有更多教授、名家动手撰写研究生教材，为建设国内一流、在国际上有影响的新四川大学做出更大的贡献．

四川大学副校长
四川大学研究生院院长　　刘应明 教授
中国科学院院士

2001年3月8日

第二版前言

本书第二版是基于第一版教材近三年来授课于工程硕士数值分析课程的教学实践及读者的有关建议，并融合当前工程数学课程建设的改革思路和教学理念，对第一版的部分内容作了调整和增添，使之更加符合工程硕士课程教学改革的需求。

新版不仅对原版教材在文字上作了全面的修订，为了扩大工程硕士的视野，还对原版的部分内容作了新的补充。例如在原版第 4 章线性方程组迭代方法的基础上增加了非线性方程组的迭代方法（Newton-Raphson 方法和 Broyden 方法）；在原版第 6 章一元插值和曲线拟合的基础上增加了多元系统分析的基本方法（多元线性回归，逐步回归和曲面拟合等）；根据 Matlab 软件第 6.1 版对原版附录 1 Matlab 及其应用的部分内容作了全面的修订。

本书第二版的修订工作由张光澄和张雷联合执笔。我们还要感谢四川大学 2006 级和 2007 级工程硕士周末班和集中班的学员对本书第一版文字方面的修订提出了宝贵的意见。

本书注有“*”的章节为选学内容。

作者

2009 年 1 月

前 言

本书是作者多年来在为四川大学相关理工科硕士生、工程硕士生、本科生、进修生开设数值分析、数值计算方法等课程及所编讲义的基础上，对搜集整理的大量材料经过充分酝酿、反复修订而成的.

本书在课程内容的处理上遵循以下原则：

1. 突出方法，适当淡化理论. 本书主要介绍计算机上常用的数值计算方法及相关的基本概念和理论，强调学用一致，突出算法的实质和思路，不刻意追求理论的系统性和完整性，注重工程硕士生带岗参加学习的实际；在内容的处理上遵循"突出重点，循序渐进"的原则，在每一章的开头大都安排有引言，开门见山提出全章的研究主题，并对必要的基础知识作适当的补充.

2. 突出应用. 本书结合实际问题介绍求解数学模型的常用算法，并配有大量的数值计算实例，引导学生主动思考，激发学生的学习兴趣.

3. 加强算法实现的基本训练. 为了帮助初学者尽快掌握算法设计的基本技巧，对每一章中涉及的主要算法都分别介绍了两种计算方案，即手算方案（表格化的计算步骤）和程序设计方案. 通过从算法到程序设计有序而系统的训练，提高学生从数值计算到简单程序设计的能力.

全书分为两个部分：第一部分为正文，共包含 8 章内容（含习题）. 第 1 章介绍算法及其基本特点和误差的基本概念；第 2 章至第 8 章介绍工程上常用的数值计算方法以及相关的基本理论. 这部分内容的设计讲授课时数为 60 学时. 第二部分包含两个附录. 附录 Ⅰ 主要介绍当今世界上最流行的数学软件 Matlab 在数值计算方法、最优化方法以及数据处理等方面的应用；附录 Ⅱ 为习题详解和参考答案. 由于数值分析课程研究的是离散对象，具有计算复杂、实践性强等特点，因而初学者对习题的解答长期感到困惑. 为此，本书对所列习题中的一些计算题目均作了详细解答，旨在帮助初学者较快入门.

四川大学管理学院侯泽华副教授对本书的校订提出了中肯的意见，四川大学管理学院郑建国同志和四川大学 2000 届工程硕士班学员刘昌明同学对本书附录 Ⅱ 部分的习题解答做了大量的工作，在此一并致以最诚恳的谢意.

作者十分感谢四川大学研究生院、四川大学出版社以及制造工程学院有关领导对本书编写出版所给予的关心和支持.

由于作者水平有限，本书在对内容的取舍、安排、衔接、表达等方面难免存在缺点甚至错误，恳请读者批评指正.

张光澄
2003 年 6 月于四川大学数学学院

目　录

第 1 章　算法及误差分析

1.1　算法简介

1.1.1　数值分析的研究对象

《微积分学》是学习《数值分析》必备的基础课程之一，这两门课程从研究内容到研究方法既有联系又有较大的差异，对此我们作简要的比较：

微积分学(Calculus)主要研究一元函数及多元函数的微分、积分和统一两者的微分方程. 以一元函数微积分为例，一元函数微积分主要研究一元函数 $y=f(x)$ 及其微分 $\mathrm{d}f(x)$、导数 $\frac{\mathrm{d}}{\mathrm{d}x}f(x)$、不定积分 $\int f(x)\mathrm{d}x$、定积分 $\int_a^b f(x)\mathrm{d}x$、微分方程 $\frac{\mathrm{d}y}{\mathrm{d}x}=\varphi(x,y(x))$ 等.

微积分学以解析运算(或称符号运算)为其研究工具，如 $f'(x)=\lim\limits_{\Delta x\to 0}\frac{\Delta y}{\Delta x}$，$\mathrm{d}f(x)=f'(x)\mathrm{d}x$，$\int f(x)\mathrm{d}x=F(x)+C$，$\int_a^b f(x)\mathrm{d}x=F(x)\big|_a^b=F(b)-F(a)$（其中 $F'(x)=f(x)$），$\frac{\mathrm{d}y}{\mathrm{d}x}=\varphi(x,y)\xrightarrow{\text{求 } y(x)}\frac{\mathrm{d}y(x)}{\mathrm{d}x}=\varphi(x,y(x))$.

数值分析(Numerical Analysis)则主要研究数值微分、数值积分、数值代数、数值微分方程以及数据处理等等. 它的研究工具是算法(Algorithm)，包括算法设计、程序设计、误差分析以及算法上机执行等内容.

下面将简要介绍算法及其基本特点.

1.1.2　算法的基本特点

1.1.2.1　算法

算法(在算法设计时简称 Algo.)是对解题步骤的准确描述，代表解题的一系列操作指令，即从输入数据或信息，经过有限步操作获得输出数据或信息.

从图 1－1 中可以看出算法在求解数学模型的过程中所起到的重要作用.

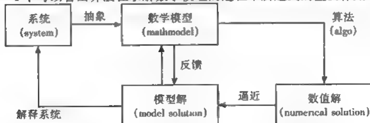


图 1－1

随着现代电子计算机和计算技术的迅速发展，算法的内容更加丰富，大致可分为如下几种类型.

1)数值型算法

数值型算法包括：数据处理，即插值、拟合、回归、列数据分析等；数值微积分，即数值微分、数值积分；数值代数，即线性方程组及非线性方程的数值解，矩阵特征值及特征向量的数值计算；数值微分方程，即微分方程的初值问题及边值问题的数值解法.

2)非数值型算法

从数值计算扩大到信息处理、数据传输，出现了相应的非数值型算法，如排序、查找、屏幕操作、图形操作、菜单设计等等.

3)密钥技术(加密、释密算法)

随着网络技术的发展，信息安全技术被提上了议事日程，电子签名、数据库存取控制、工作站执行控制、服务器存取控制等都大量使用加密、释密等算法.

1.1.2.2　算法的特点

(1)有效性.算法给出的每一步操作都有明确的指令，能在计算机环境下执行，不允许产生二义性.

例 1　设一元二次方程 $ax^2+bx+c=0$，用笛卡儿(Descartes)公式求根，即

$$x_{1,2}=\frac{-b\pm\sqrt{b^2-4ac}}{2a},\quad \begin{cases} b^2-4ac\geqslant 0 & (\text{二实根}),\\ b^2-4ac<0 & (\text{二复根}).\end{cases}$$

当 $b^2-4ac\geqslant 0$ 时，$x_{1,2}$ 表示二实根；当 $b^2-4ac<0$ 时，$x_{1,2}$ 表示二复根.为避免产生二义性，算法采用分支结构.

Algo. 1.1(分支结构)

1°　输入 a,b,c.

2°　计算判别式 $b^2-4ac\Rightarrow p$.

3°　$p\geqslant 0$?

是，$\dfrac{(-b+\sqrt{p})}{2a}\Rightarrow x_1$，$\dfrac{(-b-\sqrt{p})}{2a}\Rightarrow x_2$；

否，$\dfrac{(-b+i\sqrt{|p|})}{2a}\Rightarrow x_1$，$\dfrac{(-b-i\sqrt{|p|})}{2a}\Rightarrow x_2$.

4°　输出 x_1,x_2.

(2)有穷性.任何算法都必须在有限步内完成，对于无穷过程实施有限化，其典型的方法是截断和离散.

例 2(截断)　计算 $\sin x$ 的值，$x\in(0,\frac{\pi}{4})$.

解　据泰勒(Taylor)展式，将 $\sin x$ 在 $x=0$ 处展开，即

$$\sin x=x-\frac{x^3}{3!}+\frac{x^5}{5!}-\frac{x^7}{7!}+\cdots+(-1)^n\frac{x^{2n+1}}{(2n+1)!}+\cdots \tag{1-1}$$

其余项

$$R_n(x)=(-1)^{n+1}\frac{x^{2n+3}}{(2n+3)!}\cos\zeta,\quad \zeta=\theta x,\quad \theta\in(0,1). \tag{1-2}$$

计算 sin0.5，设在式(1-1)中取 $n=3$，则

$$\sin 0.5 \approx 0.5 - \frac{0.5^3}{3!} + \frac{0.5^5}{5!} - \frac{0.5^7}{7!} = 0.479\ 425\ 533,$$

而
$$R_3(x) = (-1)^4 \frac{x^9}{9!}\cos\zeta, \quad \zeta \in \left(0, \frac{\pi}{4}\right),$$

$$|R_3(0.5)| \leqslant \frac{\left(\frac{\pi}{4}\right)^9}{9!} = 3.13\times10^{-7} = 0.313\times10^{-6} < 0.5\times10^{-6}.$$

取 8 位小数，则 $\sin 0.5 \approx 0.479\ 425\ 53$. 这是因为精确值 $\sin 0.5$、近似值 0.479 425 53 及绝对误差限 0.5×10^{-8} 之间满足如下不等式

$$|\sin 0.5 - 0.479\ 425\ 53| < \frac{1}{2}\times10^{-8}.$$

因此所计算出的近似值准确至第 8 位小数.

例 3（离散） 在微积分中，定积分使用牛顿－莱布尼兹（Newton－Leibniz）公式

$$\int_a^b f(x)\mathrm{d}x = F(x)\Big|_a^b = F(b) - F(a).$$

但在相当多的情况下，原函数 $F(x)$ 的解析式不能用初等函数表出. 例如 $\int_0^1 \frac{\sin x}{x}\mathrm{d}x$（正弦积分）不存在初等函数 $F(x)$ 使得 $F'(x) = \frac{\sin x}{x}$；又如道森（Dawson）积分 $\varphi(x) = \mathrm{e}^{-x^2}\int_0^x \mathrm{e}^{t^2}\mathrm{d}t$ 中的积分 $\int_0^x \mathrm{e}^{t^2}\mathrm{d}t$，仍然不存在初等函数 $F(t)$ 使得 $F'(t) = \mathrm{e}^{t^2}$.

解决上述求积问题的办法是将连续问题离散化，即用数值求积的方法.

将积分区间 $[a,b]$ 分成 n 个部分，如图 1－2 所示. 令

$$\int_a^b f(x)\mathrm{d}x \approx \sum_{i=0}^{n} A_i f(x_i). \tag{1-3}$$

式中 $f(x_i)$ 是分点 x_i 处的函数值，A_i 称为求积系数.

我们称式（1－3）为数值求积公式，它将一个无穷过程转化为有穷过程，即通过有限步完成积分计算. 如何确定求积系数 A_i，我们将在第 7 章中详细介绍.

图 1－2

（3）复杂性（计算复杂性）. 算法的计算复杂性是指计算代价的估计，它包括估计存储量（内存空间）和估计计算量（计算时间）两部分，统称为时空代价的估计.

例 4 秦九韶算法：设 n 次多项式为

$$p_n(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0,$$

试计算 $p_n(x)$ 的值，并估计所用乘法的次数.

解 常规算法(逐项计算)：

$$\left.\begin{array}{ll} 1\text{ 次项} & 1\text{ 次乘法} \\ 2\text{ 次项} & 2\text{ 次乘法} \\ \vdots & \vdots \\ n\text{ 次项} & n\text{ 次乘法} \end{array}\right\} \text{共计乘法次数为 } 1+2+3+\cdots+n=\frac{n(n+1)}{2}.$$

秦氏算法(递推算法)：

设 $n=4$,则

$$\begin{aligned} p_4(x) &= a_4x^4+a_3x^3+a_2x^2+a_1x+a_0 \\ &= \{[(a_4x+a_3)x+a_2]x+a_1\}x+a_0, \end{aligned}$$

$$a_4x+a_3\Rightarrow s_1, \quad s_1x+a_2\Rightarrow s_2, \quad s_2x+a_1\Rightarrow s_3, \quad s_3x+a_0\Rightarrow s_4=p_4(x).$$

若计算一个 4 次多项式,则 $p_4(x)\begin{cases}\text{常规算法:10 次乘法,}\\ \text{秦氏算法:4 次乘法;}\end{cases}$

若计算一个 n 次多项式,则 $p_n(x)\begin{cases}\text{常规算法:}\dfrac{n(n+1)}{2}\text{次乘法,}\\ \text{秦氏算法:}n\text{ 次乘法.}\end{cases}$

可见当 n 很大时,秦氏算法的优越性尤其显著. 该算法采用循环结构.

Algo. 1.2(循环结构)

1° 输入 $a_i(i=0,1,\cdots,n),x,n$.

2° $a_n\Rightarrow s_0$, $k=1$.

3° $s_{k-1}x+a_{n-k}\Rightarrow s_k$.

4° $k\leqslant n$?

是,$k+1\Rightarrow k$, 转 3°;

否,输出 s_k, stop.

为了节省工作单元,上述算法中的数组 s(其下标量分别为 $s_0,s_1,\cdots,s_n$)可改用动态存储方式,只需引入一个工作单元 s.

Algo. 1.3(动态存储)

1° 输入 $a_i(i=0,1,\cdots,n),x,n$.

2° $a_nx\Rightarrow s$, $n\Rightarrow k$.

3° $x(a_{k-1}+s)\Rightarrow s$.

4° $k>2$?

是,$k-1\Rightarrow k$,转 3°;

否,输出 $s+a_{k-1}$, stop.

注 秦九韶,南宋时期我国杰出的数学家,于公元 1247 年写成传世名著《数书九章》,美国哈佛大学科学史家萨顿(Sarton)称秦氏是“他那个民族,他那个时代,并且确实也是所有时代最伟大的数学家之一”.

1.2 误差分析

1.2.1 误差的来源

在运用数学方法解决实际问题的过程中，每一步都可能带来误差.

(1)模型误差. 在建立数学模型时，往往要忽略很多次要因素，把模型“简单化”、“理想化”，这时模型就与真实背景有了差距，即带人了误差.

(2)测量误差. 数学模型中的已知参数，多数是通过测量得到. 而测量受工具、方法、观察者的主观因素、不可预料的随机干扰等影响，必然带入误差.

(3)舍人误差. 计算中常会遇到无理数或无穷有理数，如 π，e，$\sqrt{2}$等无理数(不尽不循环的数)，$\frac{1}{3}$等无穷有理数(不尽循环的数). 由于计算机只能处理有限数位的小数运算，因此对于上述数据或中间结果都必须进行修约，这必然产生修约误差.

(4)方法误差(截断、离散所致). 求解数学模型的数值计算方法绝大多是近似公式，这种简化带人的误差称为方法误差，它是本课程研究的主要对象.

(5)初值误差. 本课程中所涉及到的迭代公式、递推公式均需要选取初始值，而初始数据不可避免地会带有误差，我们称之为初值误差或初始扰动. 初值误差对算法的影响，即所谓算法稳定性问题，也是本课程的研究对象之一. 稳定算法，即微小的初始扰动对计算结果影响不大；而非稳定算法，即微小的初始扰动对计算结果可能影响很大.

例5 计算含参数的积分

$$I_n=\int_0^1 \frac{x^n}{x+5}\mathrm{d}x, \quad n=0,1,\cdots,8.$$

解
$$I_n+5I_{n-1}=\int_0^1 \frac{x^n+5x^{n-1}}{x+5}\mathrm{d}x=\int_0^1 x^{n-1}\mathrm{d}x=\frac{1}{n}. \qquad (1-4)$$

Algo. 1.4(正向递推)

据式(1.4)得递推公式

$$\begin{cases} I_n=\dfrac{1}{n}-5I_{n-1}, \quad n=1,2,\cdots,8, \\ I_0=\displaystyle\int_0^1 \frac{1}{x+5}\mathrm{d}x=\ln(x+5)\mid_0^1=\ln\frac{6}{5}=\ln 1.2=0.182\,321\,556. \end{cases}$$

取 $I_0\approx0.182\,3$(保留4位小数)，计算结果用I_n表示，其数值如下：

$$I_1=0.088\,5, \quad I_2=0.057\,5, \quad I_3=0.045\,8, \quad I_4=0.021\,0,$$
$$I_5=0.095\,0, \quad I_6=-0.308\,3, \quad I_7=1.684\,4, \quad I_8=-8.297\,0.$$

通过理论分析可以证明，题中的积分 I_n 满足以下规律：

$$对\ \forall n, 0<I_n<1(n=1,2,\cdots,8)，且当\ n\nearrow, I_n\searrow.$$

显然 Algo. 1.4 与理论分析相悖. 其原因是 Algo. 1.4 为非稳定算法，初值 $I_0\approx0.182\,3$ 的微小扰动将造成此后的计算失真.

Algo. 1.5(逆向递推——改进算法)

据式(1 4)得逆向递推公式

$$\begin{cases} I_{n-1}=\frac{1}{5}(\frac{1}{n}-I_n), & n=8,7,\cdots,1, \\ I_8=\int_0^1 \frac{x^8}{x+5}dx=0.018\,836\,92. \end{cases}$$

取$I_8=0.018\,84$(保留 4 位有效数字),计算结果如下:

$I_7=0.021\,23$, $I_6=0.024\,33$, $I_5=0.028\,47$, $I_4=0.034\,31$,

$I_3=0.043\,14$, $I_2=0.058\,04$, $I_1=0.088\,39$, $I_0=0.182\,3$.

显然,Algo. 1.5 的计算结果与理论分析一致,即

$$0<I_n<1(n=8,7,6,\cdots,1,0),且当 n\nearrow, I_n\searrow.$$

以上计算结果说明,Algo. 1.5 为稳定算法,初值$I_8=0.018\,84$ 的微小扰动不影响其后的计算值.

1.2.2 误差的基本概念

人们常用绝对误差或有效数字位数来描述一个近似值的准确程度,由于这些概念早为读者所熟悉,本书只作简要叙述.

1)绝对误差与绝对误差限

设 x^* 是准确值 x 的一个近似值,则称 $x-x^*$ 为近似值 x^* 的绝对误差,并用 $e^*(x)$表示,即

$$e^*(x)=x-x^*. \tag{1-5}$$

绝对误差 $e^*(x)$一般无法计算,只能估计出它的绝对值的一个上限,即求一个正数 ε^*,使得

$$|e^*(x)|=|x-x^*|\leqslant\varepsilon^*. \tag{1-6}$$

满足不等式(1-6)的正数 ε^* 称为近似值 x^* 的绝对误差限,简称为误差限.

2)有效数字与有效数字位数

设近似值 x^* 某位数的半个单位是它的误差限,而且从该位数字到 x^* 最左边的那个非零数字共有 n 位(见图 1-3),那么我们把这 n 位数字都称为有效数字,并称 x^* 具有 n 位有效数字.

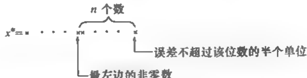


图 1-3

一个近似值,它的有效数字的位数与它的绝对误差限之间有紧密的关系.

(1)已知 ε^*,确定 x^* 的有效数字的个数或者确定 x^* 准确至某一数位.

[法则 1] 若 $\varepsilon^*=\frac{1}{2}\times10^{-k}$,则满足$|x^*-x|\leqslant\varepsilon^*$ 的近似数 x^* 准确至第 k 位小数,而它的有效数字的位数等于第 k 位小数至第一个非零数字的位数.

例 6 设 $\pi=3.14159265359\cdots$讨论 π 的近似值的有效位数.

解 取 π 的 2 位小数的近似值 $x^*=3.14$. 因为

$$|3.14-\pi|\leqslant\frac{1}{2}\times10^{-2},$$

即 $x=3.14$ 准确至第 2 位小数,具有 3 位有效数字.

取 π 的 3 位小数的近似值 $x^*=3.142$. 因为

$$|3.142-\pi|\leqslant\frac{1}{2}\times10^{-3},$$

即 $x^*=3.142$ 准确至第 3 位小数,具有 4 位有效数字.

(2)已知 x^* 的有效数字的位数(或准确至某一数位),确定 x^* 的绝对误差限.

[**法则 2**] 若 x^* 准确至第 k 位小数,则

$$\varepsilon^*=\frac{1}{2}\times10^{-k}.$$

在例 2 中,$\sin 0.5\approx0.47942553$,即准确至第 8 位小数,因为

$$|R_3(0.5)|=|\sin 0.5-0.47942553|\leqslant\frac{1}{2}\times10^{-8}.$$

对同一个数的近似值来说,有效数字位数越多,其绝对误差越小;而绝对误差越小,有效数字的位数有可能越多.

注 *在南北朝时期,我国伟大的数学家祖冲之以"更开密法"获得圆周率 π 的"盈朒二限",即*

$$3.1415926<\pi<3.1415927.$$

"盈朒二限"的平均值为 3.14159265,已准确至第 8 位小数,这是当时世界上的最佳结果. 祖冲之用几何方法将圆周率表示成密率 $\frac{355}{113}$ 更是世界数学史上的卓越成就. 一千多年后德国数学家奥托(Otto)等人才重新获得这一结果.

习题 1

(略)

第2章 非线性方程的数值解法

非线性方程求根是工程计算和科学研究中最常见的数学问题之一. 但是,除极少数简单方程外,通常都难以获得求根的简单易用的显式表达式. 因此,需要研究求根的近似值的方法.

2.1 引言

2.1.1 一元非线性方程求根

本章讨论的非线性方程 $f(x)=0$ 通常包括两种类型:一类是代数多项式方程,即 $f(x)$为代数多项式

$$f(x)=\sum_{i=0}^{n}a_ix^i=a_0+a_1x+\cdots+a_nx^n;$$

另一类是超越方程,即 $f(x)$为超越函数. 例如

$$f(x)=xe^x-1,$$

$$f(x)=e^x+32.5x-\sec\sqrt{1.79x}.$$

所谓方程求根,即是求 $x^*\in[a,b]$,使得$f(x^*)\equiv0$,或者求 $x^*\in D\subset R^2$,使得 $f(x^*)\equiv0$. 前者x^*为 $f(x)=0$ 的实根;后者 x^*为 $f(x)=0$ 的复根.

例 1 五次代数方程 $x^5+2x+1=0$ 有五个根,其中一个实根,四个复根,即

$$x_1=-0.486\,389,\quad x_{2,3}=-0.701\,874\pm0.879\,697\mathrm{i},\quad x_{4,5}=0.945\,068\pm0.854\,518\mathrm{i}.$$

例 2 三次代数方程 $x^3-3x+1=0$ 有三个根,即

$$x_1\in(-2,-1),\quad x_2\in(0,1),\quad x_3\in(1,2).$$

例 3 超越方程 $xe^x-1=0$ 在$(-1,+\infty)$内只有一个实根.

例 4 超越方程 $x-\ln x-2=0$ 在$[2,+\infty)$内只有一个实根.

对于非线性方程 $f(x)=0$,判别根的存在性通常运用微积分中的介值定理.

定理 2.1 设 $f(x)\in C_{[a,b]}$,且满足 $f(a)\cdot f(b)<0$,则 $f(x)=0$ 在$[a,b]$内至少有一实根;若 $f'(x)\geqslant0$ 或 $f'(x)\leqslant0,x\in[a,b]$,则 $f(x)=0$ 在$[a,b]$上仅有一单实根或重实根(如图2-1).

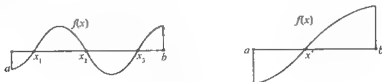


图2-1

上述定理是充分条件,不一定必要.不满足定理条件时会出现异常情况.

1)满足连续条件,但不满足变号条件

在图 2-2(A)中,$f(x)$不满足变号条件,$f(x)=0$ 无实根;在图 2-2(B)中,$f(x)$不满足变号条件,但 $f(x)=0$ 却有两个相同的实根(二重根).在图 2-3 中,$f(x)$不满足变号条件,但 $f(x)=0$ 却有两个相异的实根.

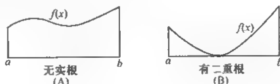


图 2-2

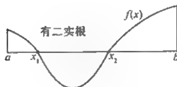


图 2-3

重根的判别条件是:

若 $f(x)=(x-x^*)^m\varphi(x)$且 $\varphi(x^*)\neq 0$,当 $m=1$ 时,称 x^* 为单根;当 $m>1$ 时,称 x^* 为 m 重实根.

2)不满足连续条件

设 $$f(x)=e^x+32.5x-\sec\sqrt{1.79x},\quad x\geqslant 0.$$

如图 2-4 所示,$f(x)$在 $x=\dfrac{\pi^2}{4\times 1.79}$处间断(即当$\sqrt{1.79x}=\dfrac{\pi}{2}$,$x=1.378\,44$ 为一间断点),但 $f(x)=0$ 在$[0,+\infty)$上有二实根:$x_1=0$,$x_2=1.341\,69$.

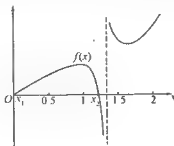


图 2-4

下面我们用定理 2.1 来检查例 2、例 3 及例 4 中的三个方程.

例 2 中,$f(x)=x^3-3x+1$ 在$(-2,-1)$,$(0,1)$,$(1,2)$内均变号,在$(-\infty,+\infty)$上连续,故由定理 2.1,$f(x)=0$ 有三个实根,且分别属于上述三个区间.

例 3 中，$f(x)=xe^x-1$. 因为 $f(0)<0, f(1)>0$，故在(0,1)内至少有一实根，且

$$f'(x)=(1+x)e^x>0, \quad x\in(-1,+\infty),$$

故 $f(x)=0$ 在 $(-1,+\infty)$ 内只有一个实根.

例 4 中，$f(x)=x-\ln x-2$. 因为 $f(2)<0, f(4)>0$，故在(2,4)内至少有一实根，且

$$f'(x)=\frac{x-1}{x}>0, \quad x\in[2,+\infty),$$

故 $f(x)=0$ 在 $[2,+\infty)$ 内仅有一实根.

2.1.2 求根的精确化方法

本章研究的重点是求根的精确化方法，根的精确化方法通常有以下两类.

1)区间搜索法

设方程 $f(x)=0$ 的一个实根 $x^*\in[a,b]$，用区间搜索法产生包含 x^* 的区间套(一系列区间)，即

$$[a,b]\supset[a_1,b_1]\supset\cdots\supset[a_k,b_k]\supset\cdots$$

设每一个区间的中点分别为

$$x_0=\frac{a+b}{2}, x_1=\frac{a_1+b_1}{2}, x_2=\frac{a_2+b_2}{2}, \cdots, x_k=\frac{a_k+b_k}{2}\cdots$$

当 k 充分大时，令 $x^*\approx x_k$.

2)迭代法

设方程 $f(x)=0$ 的一个实根 $x^*\in[a,b]$，取初始点 $x_0\in[a,b]$ 或 $x_0\in U(x^*,\delta)$，用迭代公式产生点列 $\{x_k\}(x=0,1,2,\cdots)$，当 k 充分大时，令 $x^*\approx x_k$.

2.2 二分法

二分法是一种最简单的区间搜索法，适用于求有限区间内的单实根或奇重实根.

2.2.1 基本二分法

2.2.1.1 问题陈述

设 $f(x)=0$，且

(1) $f(x)\in C_{[a,b]}$；

(2) $f(a)\cdot f(b)<0$；

(3) $f'(x)\geqslant 0$ 或 $f'(x)\leqslant 0, x\in[a,b]$.

求 $f(x)=0$ 在 $[a,b]$ 内仅有的一个实根.

2.2.1.2 求解方法

不失一般性，设 $f(a)<0, f(b)>0$(如图 2-5).

(1)将区间 $[a,b]$ 二等分，得其子区间 $[a_1,b_1]$，令 $[a,b]$ 的长为 $\Delta=b-a$，$[a_1,b_1]$ 的长为 $\Delta_1=\frac{b-a}{2}$. 确定 $[a_1,b_1]$ 的具体做法是：

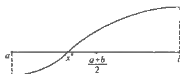


图 2-5

第一步，取$[a,b]$的中点$x_0=\frac{a+b}{2}$.

第二步，检查：$f(x_0)=0$?

是，$x_0 \Rightarrow x^*$， Stop.

否，再检查：$f(x)>0$?

是，$a \Rightarrow a_1, x \Rightarrow b_1$，得$[a_1,b_1]=\left[a,\frac{a+b}{2}\right]$；

否，$x \Rightarrow a_1, b \Rightarrow b_1$，得$[a_1,b_1]=\left[\frac{a+b}{2},b\right]$.

(2)将$[a_1,b_1]$再次二等分得其子区间$[a_2,b_2]$，令其长为$\Delta_2=\frac{b-a}{2^2}$. 仍取$[a_1,b_1]$的中点$x_1=\frac{a_1+b_1}{2}$.

重复(1)中第二步的做法，得$[a_2,b_2]$，即$[a_2,b_2]=\left[a_1,\frac{a_1+b_1}{2}\right]$或者$\left[\frac{a_1+b_1}{2},b_1\right]$.

(3)设已搜索k次，由此得一区间套

$$[a,b]\supset[a_1,b_1]\supset\cdots\supset[a_k,b_k],$$

其中，区间$[a_k,b_k]$的长$\Delta_k=\frac{b-a}{2^k}$. $x^*\in[a_k,b_k]$即为$f(x)=0$在$[a,b]$内的一个实根，当k充分大时，令$x_k=\frac{1}{2}(a_k+b_k)\approx x^*$.

2.2.1.3 误差估计

因为$f(x)=0$的一个实根$x^*\in[a_k,b_k]$，如图2－6所示，所以

$$|x_k-x^*|\leqslant\frac{1}{2}\Delta_k=\frac{b-a}{2^{k+1}}.$$



图2－6

若给定x_k的绝对误差限为ε，则$|x_k-x^*|\leqslant\varepsilon$，由此可确定二分次数$k$.

令$\varepsilon=\frac{b-a}{2^{k+1}}$，由$2^k=\frac{b-a}{2\varepsilon}$，解得

$$k=\left[\frac{\lg\frac{b-a}{2\varepsilon}}{\lg 2}\right].$$

注 k取大于括号中数的最小整数.

2.2.1.4 手算方案

例5 求$f(x)=x^3-3x+1=0$在$[0,1]$内的一个实根，要求精确至3位小数.

解 第一步，确定二分次数.

因为 $$\varepsilon=\frac{1}{2}\times10^{-3},\quad b-a=1,$$

所以 $$k=\left[\frac{3}{\lg 2}\right]=[9.97]\approx10.$$

即是说，所求根欲达到给定的精度，需要二分 10 次.

第二步，列表计算(见表 2-1，保留 4 位小数).

表2-1

k	$f(x)$	存根区间	k	$f(x)$	存根区间
	$f(0)=1>0$		5	$f(0.3438)>0$	$[0.3438, 0.375]$
	$f(1)=-1<0$	$[0,1]$	6	$f(0.3594)<0$	$[0.3438, 0.3594]$
1	$f(0.5)<0$	$[0,0.5]$	7	$f(0.3516)<0$	$[0.3438, 0.3516]$
2	$f(0.25)>0$	$[0.25, 0.5]$	8	$f(0.3477)<0$	$[0.3438, 0.3477]$
3	$f(0.375)<0$	$[0.25, 0.375]$	9	$f(0.3458)>0$	$[0.3458, 0.3477]$
4	$f(0.3125)>0$	$[0.3125, 0.375]$	10	$f(0.3468)>0$	$[0.3468, 0.3477]$

所以 $x^* \approx x_{10}=\frac{1}{2}(0.3468+0.3477)=0.3473\approx 0.347$.

2.2.2 二分法算法设计

设 $f(x)=0$，$f(x)$在$[a,b]$上连续，且在区间两端点变号，欲求该方程的实根通常采用两个步骤.

第一步，用逐步扫描法搜索较小的存根区间. 计算时可以从左端点 $x=a$ 出发，按选定的步长 h 一步一步地向右跨，每跨一步检查是否是变号区间. 若是，则该步内必有实根. 步长的选择视具体情况而定. 显然在扫描过程中步长 h 选择得足够小，就可以得到满足一定精度的根的近似值，但这种方法一般不可取，通常只要搜索出包含根的一个较小的分布区间就可以了. 具体做法见算法 Algo. 2. 1：

Algo. 2.1(逐步扫描法)

1° 输入 a,b,h.

2° 1→flag.

3° $f(a)\to s$.

4° $a+h\leqslant b$ 和 $s\cdot f(a+h)>0$?

是，$a+h\to a$，转 3°；

否，转 5°.

5° $a+h>b$?

是，0→flag，转 6°；

否，$a+h\to b$，转 6°.

6° flag=1?

是，输出 a,b， stop；

否，输出 flag， stop.

第二步，在变号区间内，若 $f(x)$满足单调性，则用基本二分法求根，见 Algo. 2. 2.

Algo. 2.2(基本二分法)

1° 输入 a,b,ε.

2° $\frac{a+b}{2}\to x$.

3° $f(x)=0$?

是，输出 x， stop；

否，转 4°.

4° $f(x)\cdot f(a)<0$?

是，$x\rightarrow b$；

否，$x\rightarrow a$.

5° $|b-a|<\varepsilon$?

是，$\frac{a+b}{2}\rightarrow x$，输出 x， stop；

否，$\frac{a+b}{2}\rightarrow x$，转 3°.

二分法方法简单，但收敛速度慢.例如求方程 $f(x)=xe^x-1=0$ 在$[0,1]$内的一个实根，若取 $\varepsilon=\frac{1}{2}\times10^{-5}$，经 17 次二分($k=17$)得到

$$[a_{17},b_{17}]=[0.567\,139,0.567\,146].$$

令
$$x^*\approx x_{17}=\frac{1}{2}(a_{17}+b_{17})=0.567\,142\,5\approx0.567\,14.$$

二分法适于求单实根，不宜于求偶重根，更不能求复根.

2.3 迭代法

如本章引言中所述，迭代法是一种逐次逼近的方法，它是求解方程或方程组的一种基本方法，在科学与工程计算中经常使用.下面将介绍几种常用的迭代方法，并且讨论迭代方法的收敛性、收敛阶(速度)以及误差估计.

2.3.1 简单迭代法

2.3.1.1 迭代公式

通过代数恒等变形，将方程 $f(x)=0$ 化成与之等价的方程 $x=\varphi(x)$.

$x=\varphi(x)$是一个隐式方程，它只是形式上解出了 x.若将隐式 $x=\varphi(x)$ 显式化，即令

$$x_{k+1}=\varphi(x_k), \tag{2-1}$$

给定初值 $x_0\in[a,b]$，重复代入式(2-1)，产生点列$\{x_k\}_{k=0,1,2,\cdots}$.

令$\lim\limits_{k\to\infty}x_k=x^*$，只要 $\varphi(x)$连续，就有

$$\lim_{k\to\infty}x_{k+1}=\lim_{k\to\infty}\varphi(x_k)=\varphi(\lim_{k\to\infty}x_k),$$

即
$$x^*=\varphi(x^*).$$

这里 x^* 是 $x=\varphi(x)$的根，亦即 $f(x)=0$ 的根.

定义 2.1 式(2-1)即 $x_{k+1}=\varphi(x_k)$称为简单迭代公式，其中 $\varphi(x)$为迭代函数.给定初值 x_0，由迭代公式(2-1)产生点列$\{x_k\}_{k=0,1,2,\cdots}$，若

$$\lim_{k\to\infty}x_k=x^*,$$

则 x^*满足 $x^*=\varphi(x^*)$，称 x^* 为 $x=\varphi(x)$的不动点，即方程 $f(x)=0$ 的根.

上述通过简单迭代公式求方程根的方法称为简单迭代法.

例 6 求 $f(x)=x^3-3x+1=0$ 的三个实根.

解 通过恒等变形,将 $f(x)=0$ 化为等价方程 $x=\varphi(x)$,有如下三种形式:

$$x=\frac{x^3+1}{3}=\varphi(x),\quad x=\frac{1}{3-x^2}=\varphi(x),\quad x=\sqrt[3]{3x-1}=\varphi(x).$$

由此构成三个迭代公式,即

$$x_{k+1}=\frac{x_k^3+1}{3},\tag{2-2}$$

$$x_{k+1}=\frac{1}{3-x_k^2},\tag{2-3}$$

$$x_{k+1}=\sqrt[3]{3x_k-1}.\tag{2-4}$$

我们知道原方程 $x^3-3x+1=0$ 在 $(-\infty,+\infty)$ 内的三个实根分别分布在 $(-2,-1)$,$(0,1)$,$(1,2)$ 三个区间内.

(1)采用迭代公式(2-2)及(2-3),取同一初值 $x_0=0.5$,计算原方程在(0,1)内的一个实根.

将迭代公式(2-2),(2-3)分别记为

$$\bar{x}_{k+1}=\frac{\bar{x}_k^3+1}{3},\quad \tilde{x}_{k+1}=\frac{1}{3-\tilde{x}_k^2}.$$

列表计算(见表 2-2,保留 6 位小数).

表 2-2

k	$\bar{x}_k$	$\tilde{x}_k$	k	$\bar{x}_k$	$\tilde{x}_k$
0	0.5	0.5	5	0.347 303	0.347 297
1	0.375	0.363 636	6	0.347 297	0.347 296
2	0.350 911	0.348 703	7	0.347 296	0.347 296
3	0.347 737	0.347 414	8	0.347 296	—
4	0.347 350	0.347 306	—	—	—

由表 2-2 可知,两个迭代公式分别通过 8 次迭代和 7 次迭代,计算结果均准确至第 6 位小数,即

$$x^*\approx\bar{x}_8=0.347\,296,\quad x^*\approx\tilde{x}_7=0.347\,296.$$

(2)采用迭代公式(2-4),即

$$x_{k+1}=\sqrt[3]{3x_k-1}.$$

分别取初值 $x_0^{(1)}=1.5$,$x_0^{(2)}=-1.5$,计算原方程在(1,2)及(-2,-1)内的两个实根(见表 2-3,保留 6 位小数).

表 2-3

k	$x_k^{(1)}$	$x_k^{(2)}$
0	1.5	−1.5
1	1.518 294	−1.765 174
2	1.526 189	−1.846 478
3	1.529 571	−1.870 022
4	1.531 015	−1.876 731
5	1.531 631	−1.878 633
6	1.531 893	−1.879 172
7	1.532 005	−1.879 325
8	1.532 053	−1.879 368
9	1.532 074	−1.879 380
10	1.532 083	−1.879 383
11	1.532 086	−1.879 385
12	1.532 088	−1.879 385
13	1.532 088	—

由表 2−3 可得 $x_1^* \approx x_{13}^{(1)} = 1.532\ 088$(准确至第 6 位小数)，$x_2^* \approx x_{12}^{(2)} = -1.879\ 385$(准确至第 6 位小数).

(3)读者自行验算以下结论：对于迭代公式(2−2)及(2−3)，如果在区间(−2，−1)或区间(1,2)内取初值，则这两个迭代公式产生的点列不收敛于该两个区间内的根. 同样，对于迭代公式(2−4)，如果在区间(0,1)内取初值，则该迭代公式所产生的点列也不收敛于该区间内的根. 换句话讲，不能用迭代公式(2−2)及(2−3)求原方程在(−2，−1)及(1,2)内的两个实根，也不能用迭代公式(2−4)求(0,1)内的一个实根.

综上所述，一个方程 $f(x)=0$ 所构成的简单迭代式 $x_{k+1}=\varphi(x_k)$ 是否收敛于该方程的根，既与迭代函数 $\varphi(x)$ 有关，也与初值 x_0 的选取有关.

2.3.1.2　迭代法的几何意义

如图 2−7 和图 2−8 所示，将方程 $f(x)=0$ 化为等价方程 $x=\varphi(x)$，从图形上看，$f(x)=0$ 的根 x^* 即是直线 $y=x$ 与曲线 $y=\varphi(x)$ 的交点(不动点)的横坐标. 在图 2−7 中，曲线 $y=\varphi(x)$ 位于直线 $y=x$ 的下方，任给初值 x_0，迭代式 $x_{k+1}=\varphi(x_k)$ 收敛于 x^*；而在图 2−8 中，曲线 $y=\varphi(x)$ 位于直线 $y=x$ 上方，任给初值 x_0，迭代式 $x_{k+1}=\varphi(x_k)$ 不收敛.

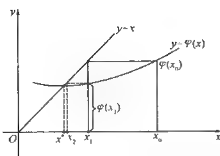


图 2—7

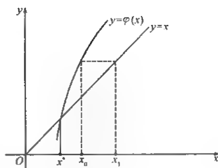


图 2—8

以上我们从几何图形上给简单迭代法作了粗略的解释，并且看到了收敛与不收敛两种情形，至于造成迭代法敛散的原因我们将在后面作进一步的分析.

2.3.1.3 算法设计

Algo. 2.3(简单迭代法)

1° 输入 x_0,ε,N(最大迭代次数).

2° $\varphi(x_0)\to x, 1\to k$.

3° $1\to$flag.

4° $k<N$?

是，$k+1\to k$，转 5°；

否，转 6°.

5° $|x-x_0|>\varepsilon$?

是，$x\to x_0,\varphi(x_0)\to x$，转 4°；

否，转 6°.

6° 若 $k\geqslant N$，$0\to$flag；

若 flag=1，则输出 x， stop；

若 flag=0，则输出 flag(算法失败).

2.3.2 加速迭代公式

当迭代公式收敛很慢时，我们可以从 $x=\varphi(x)$ 出发构造出新的迭代形式，以加快收敛速度.

2.3.2.1 松驰法(预报 校正公式)

对于给定的初值 x_k，令 $\bar{x}_{k+1}=\varphi(x_k)$ 作为第 k 次迭代的预报值，将 x_k 与 $\bar{x}_{k+1}$ 作加权平均；令 $x_{k+1}=w_k\bar{x}_{k+1}+(1-w_k)x_k$ 作第 k 次迭代的校正值，从而形成加速迭代的预报校正公式

$$\begin{cases}\bar{x}_{k+1}=\varphi(x_k),\\ x_{k+1}=w_k\bar{x}_{k+1}+(1-w_k)x_k.\end{cases}\tag{2-5}$$

其中，$w_k=\dfrac{1}{1-L}$ 称为松驰因子，且

$$L=\varphi'(x_k)\approx\varphi'(\zeta),$$
$$\zeta=x_k+\theta(x_k-x^*),\quad \theta\in(0,1).$$

容易证明,校正值 x_{k+1} 较之于预报值 $\bar{x}_{k+1}$ 更接近于 x^*. 事实上

$$\begin{aligned}\bar{x}_{k+1}-x^* &=\varphi(x_k)-\varphi(x^*)=\varphi'(\zeta)(x_k-x^*)\\ &\approx\varphi'(x_k)(x_k-x^*)=L(x_k-x^*).\end{aligned}\qquad(2-6)$$

从式(2-6)中解出 x^*,有

$$\begin{aligned}x^* &\approx\frac{1}{1-L}\bar{x}_{k+1}-\frac{L}{1-L}\bar{x}_{k+1}=\frac{1}{1-L}\bar{x}_{k+1}+\left(1-\frac{1}{1-L}\right)x_k\\ &=w_k\bar{x}_{k+1}+(1-w_k)x_k=x_{k+1}.\end{aligned}$$

2.3.2.2 埃特金(Aitken)加速迭代公式

为避免确定松驰因子 $w_k=\dfrac{1}{1-\varphi'(x_k)}$ 的麻烦,Aitken 提出两次校正的加速迭代公式

$$\begin{cases}\bar{x}_{k+1}=\varphi(x_k),\\ \tilde{x}_{k+1}=\varphi(\bar{x}_{k+1}),\\ x_{k+1}=\tilde{x}_{k+1}-\dfrac{(\tilde{x}_{k+1}-\bar{x}_{k+1})^2}{\tilde{x}_{k+1}-2\bar{x}_{k+1}+x_k}.\end{cases}\qquad(2-7)$$

容易证明,x_{k+1} 较之 $\tilde{x}_{k+1}$ 更接近 x^*. 事实上

$$\begin{aligned}x^* &=\tilde{x}_{k+1}+x^*-\tilde{x}_{k+1}\\ &=\tilde{x}_{k+1}+\varphi(x^*)-\varphi(\bar{x}_{k+1})\\ &=\tilde{x}_{k+1}+\varphi'(\zeta)(x^*-\bar{x}_{k+1}).\end{aligned}\qquad(2-8)$$

令 $\varphi'(\zeta)\approx\dfrac{\varphi(\tilde{x}_{k+1})-\varphi(\bar{x}_{k+1})}{\tilde{x}_{k+1}-\bar{x}_{k+1}}$,并将其代人式(2-8)解出 x^*,得到

$$x^*\approx\tilde{x}_{k+1}-\frac{(\tilde{x}_{k+1}-\bar{x}_{k+1})^2}{\tilde{x}_{k+1}-2\bar{x}_{k+1}+x_k}=x_{k+1}.$$

例 7 设 $x^3-3x+1=0$,试分别用加速迭代公式(2-5)及(2-7)计算方程在区间(0,1)内的一个实根.

解 (1)选用简单迭代公式(2-1)作预报,用松驰公式(2-5)进行计算,即

$$\begin{cases}\bar{x}_{k+1}=\varphi(x_k)=\dfrac{x_k^3+1}{3},\\ x_{k+1}=w_k\bar{x}_{k+1}+(1-w_k)x_k.\end{cases}$$

其中,$w_k=\dfrac{1}{1-\varphi'(x_k)}=\dfrac{1}{1-x_k^2}$.

取 $x_0=0.5$,列表计算(保留 6 位小数),其结果见表 2-4.

表 2－4

k	x_k	w_k	$\bar{x}_{k+1}$
0	0.5	1.333 333	0.375 000
1	0.333 333	1.125 000	0.345 679
2	0.347 222	1.137 091	0.347 287
3	0.347 296	1.137 158	0.347 296
4	0.347 296	—	—

由表 2－4 可得，$x^* \approx x_4 = 0.347\ 296$(准确至第 6 位小数).

(2)用 Aitken 加速公式(2－7)进行计算，即

$$\begin{cases} \bar{x}_{k+1} = \varphi(x_k) = \dfrac{x_k^3+1}{3}, \\ \tilde{x}_{k+1} = \varphi(\bar{x}_{k+1}) = \dfrac{(\bar{x}_{k+1})^3+1}{3}, \\ x_{k+1} = \tilde{x}_{k+1} - \dfrac{(\tilde{x}_{k+1}-\bar{x}_{k+1})^2}{\tilde{x}_{k+1}-2\bar{x}_{k+1}+x_k}. \end{cases}$$

仍取 $x_0 = 0.5$，列表计算(保留 6 位小数)，其结果见表 2－5.

表 2－5

k	x_k	$\bar{x}_{k+1}$	$\tilde{x}_{k+1}$
0	0.5	0.375	0.350 911
1	0.345 161	0.347 040	0.347 265
2	0.347 296	—	—

运用 Aitken 加速公式(2－7)计算，只需两次迭代便得到与前面同样的结果.

在例 6 中，我们曾建议读者验算简单迭代公式(2－2)，即 $x_{k+1} = \dfrac{x_k^3+1}{3}$ 在区间(1,2)内取初值时不收敛.

现若取初值 $x_0 = 1.5 \in (1,2)$，用简单迭代公式(2－2)作预报，用 Aitken 加速公式(2－7)计算，经 5 次迭代可求得原方程根的一个较好的近似值 x_5，即

$$x^* \approx x_5 = 1.532\ 088 \text{(准确至第 6 位小数)}.$$

例 8 求方程 $xe^x - 1 = 0$ 在区间(0,1)内的一个实根.

解 (1)用简单迭代法构造迭代公式，即

$$x_{k+1} = e^{-x_k}.$$

取初值 $x_0 = 0.5$，迭代 23 次得 $x_{23} = 0.567\ 143$(准确至第 6 位小数).

(2)用加速迭代公式(2－5)，即

$$\begin{cases}\bar{x}_{k+1}=e^{-x_k},\\ x_{k+1}=w_k\bar{x}_{k+1}+(1-w_k)x_k.\end{cases}$$

其中，$w_0=\dfrac{1}{1-\varphi'(0.5)}=\dfrac{1}{1+e^{-0.5}}=0.622\ 459.$

取 $x_0=0.5$，迭代 3 次得 $x_3=0.567\ 143.$

2.3.3 牛顿(Newton)迭代法

2.3.3.1 Newton 迭代公式

简单迭代是将非线性方程 $f(x)=0$ 通过代数恒等变形，将原方程化成等价方程 $x=\varphi(x)$，从而形成迭代式 $x_{k+1}=\varphi(x_k)$.

Newton 迭代法则是将非线性方程 $f(x)=0$ 在 x_0 点展开，即

$$f(x)=f(x_0)+f'(x_0)(x-x_0)+\frac{f''(\xi)}{2!}(x-x_0)^2,$$

并令
$$p(x)=f(x_0)+f'(x_0)(x-x_0).$$

用线性方程 $p(x)=0$ 近似替代非线性方程 $f(x)=0$，再从 $p(x)=0$ 中解得

$$x=x_0-\frac{f(x_0)}{f'(x_0)}.$$

并令 $x_1=x_0-\dfrac{f(x_0)}{f'(x_0)}$作为 $f(x)=0$ 的根的第一级近似值. 一般地，记

$$x_{k+1}=x_k-\frac{f(x_k)}{f'(x_k)}, \tag{2-9}$$

作为方程 $f(x)=0$ 的根的第 $k+1$ 级近似值.

我们称公式(2－9)为 Newton 迭代公式，其中 x_k 为第 k 次迭代的初值.

Newton 迭代公式(2－9)也可以由加速迭代公式(2－5)导出. 设 $f(x)=0$ 的等价方程为 $x=x+f(x)$，由此构成简单迭代式 $x_{k+1}=x_k+f(x_k)$，按式(2－5)写出加速迭代公式为

$$\begin{cases}\bar{x}_{k+1}=x_k+f(x_k)=\varphi(x_k),\\ x_{k+1}=w_k\bar{x}_{k+1}+(1-w_k)x_k.\end{cases} \tag{2-10}$$

其中，$w_k=\dfrac{1}{1-\varphi'(x_k)}=-\dfrac{1}{f'(x_k)}.$

将 $w_k=-\dfrac{1}{f'(x_k)}$代入式(2－10)的第二个式子，即得

$$x_{k+1}=x_k-\frac{f(x_k)}{f'(x_k)}.$$

这就是 Newton 迭代公式.

例 9 用 Newton 迭代法求方程 $xe^x-1=0$ 在区间(0,1)内的一个实根.

解 据式(2－9)，本例的 Newton 迭代公式为

$$x_{k+1}=x_k-\frac{x_ke^{x_k}-1}{e^{x_k}(x_k+1)}=x_k-\frac{x_k-e^{-x_k}}{x_k+1}.$$

取初值 $x_0=0.5$，列表计算(保留 6 位小数)，其结果见表 2－6.

表 2－6

k	x_k	k	x_k
0	0.5	3	0.567 143
1	0.571 020	4	0.567 143
2	0.567 156	—	—

由表 2－6 可得 $x^* \approx x_4=0.567143$(准确至第 6 位小数).

与例 8 中的简单迭代法比较，取同一初值 $x_0=0.5$，简单迭代法经 23 次迭代得 $x_{23}=0.567143$，用 Newton 迭代法经 4 次迭代即得 $x_4=0.567143$.

例 10 用 Newton 迭代法求方程 $x^3-3x+1=0$ 的全部实根.

解 运用 Newton 迭代公式有

$$x_{k+1}=x_k-\frac{x_k^3-3x_k+1}{3x_k^2-3}.$$

分别取初值 $x_0^{(1)}=0.5, x_0^{(2)}=1.5, x_0^{(3)}=-1.5$，并列表计算(保留 6 位小数)，其结果见表 2－7.

表 2－7

k	$x_k^{(1)}$	$x_k^{(2)}$	$x_k^{(3)}$
0	0.5	1.5	−1.5
1	0.333 333	1.533 333	−2.066 667
2	0.347 222	1.532 091	−1.900 876
3	0.347 296	1.532 089	−1.879 720
4	0.347 296	1.532 089	−1.879 385
5	—	—	−1.879 385
6	—	—	—
7	—	—	—

由表 2－7 可得

$$x_1^* \approx x_4^{(1)}=0.347296,\quad x_2^* \approx x_4^{(2)}=1.532089,\quad x_3^* \approx x_5^{(3)}= \ 1.879385.$$

将上述结果与例 6 中的简单迭代法作比较，显然用 Newton 迭代法能以更快的速度收敛于原方程的三个实根. 同时我们还看到，用一个 Newton 迭代式可同时求得原方程的三个实根，而用简单迭代法需用不同的迭代式才能求三个实根.

2.3.3.2 Newton 迭代法的几何意义

如图 2－9 所示，方程 $f(x)=0$ 的解就是曲线 $y=f(x)$ 与 x 轴交点的横坐标 x^*. 设 x_k 为初值，过点 $(x_k, f(x_k))$ 作 $y=f(x)$ 的切线，则切线方程为

$$y-f(x_k)=f'(x_k)(x-x_k).$$

它与 x 轴的交点横坐标(令 $y=0$)为

$$x=x_k-\frac{f(x_k)}{f'(x_k)}.$$

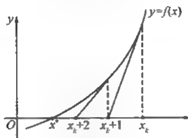


图 2～9

令 $x_{k+1}=x_k-\dfrac{f(x_k)}{f'(x_k)}$,即是第 k 次迭代的迭代点.

2.3.3.3 算法设计

Algo. 2.4(Newton 迭代法)

1° 输入 x_0,ε,N.

2° 计算 $f(x_0),f'(x_0)$.

3° $x_0-\dfrac{f(x_0)}{f'(x_0)}\to x$.

4° $1\to k,1\to \text{flag}$.

5° $k<N$?

是,$k+1\to k$,转 6°;

否,转 7°.

6° $|x-x_0|>\varepsilon$?

是,$x\to x_0,x_0-\dfrac{f(x_0)}{f'(x_0)}\to x$,转 5°;

否,转 8°.

7° 若 $k\geqslant n,0\to \text{flag}$.

8° 若 flag=1,则输出 x;

若 flag=0,则输出 flag(算法失败).

2.4 迭代法收敛性分析

下面给出简单迭代法(包含 Newton 迭代法)收敛性定义以及判别迭代法收敛性的两个重要定理.

2.4.1 收敛性定义

定义 2.2(非局部收敛或全局收敛) 对 $\forall x_0\in[a,b]$,由迭代式 $x_{k+1}=\varphi(x_k)$所产生的点列收敛,即$\lim\limits_{k\to\infty}x_k=x^*$,则称 $x_{k+1}=\varphi(x_k)$在$[a,b]$上收敛于 x^*.

定义 2.3(局部收敛)　若存在 x^* 的某邻域 $R=\{x \mid |x-x^*|<\delta\}$，对 $\forall x_0\in R$，由 $x_{k+1}=\varphi(x_k)$ 所产生的点列收敛，则称 $x_{k+1}=\varphi(x_k)$ 在 R 上收敛于 x^*.

2.4.2　收敛性判别条件

定理 2.2(非局部收敛判别定理)　设 $x_{k+1}=\varphi(x_k)$，其中 $\varphi(x)$ 满足：

(1)闭合条件，即对 $\forall x\in[a,b]$，$\varphi(x)\in[a,b]$，

(2)压缩条件，即对 $\forall x\in[a,b]$，$\exists\, 0<q<1$，使得 $|\varphi'(x)|\leqslant q$，

则有如下结论：

(1) $x_{k+1}=\varphi(x_k)$ 在 $[a,b]$ 上惟一地收敛于 x^*，且 $x^*=\varphi(x^*)$，

(2) $|x_k-x^*|\leqslant\dfrac{1}{1-q}|x_{k+1}-x_k|$.　　(2-11)

上述定理称为存在-收敛性定理，又称为康托诺维奇(Kantorovich)分析.

证明　首先证明收敛性.

因为
$$x_{k+1}-x^*=\varphi(x_k)-\varphi(x^*)=\varphi'(\zeta)(x_k-x^*),$$
故
$$|x_{k+1}-x^*|=|\varphi'(\zeta)|\cdot|x_k-x^*|\leqslant q|x_k-x^*|. \tag{2-12}$$

反复递推式(2-12)得
$$\begin{aligned}|x_k-x^*|&\leqslant q|x_{k-1}-x^*|\\&\leqslant q^2|x_{k-2}-x^*|\\&\;\;\vdots\\&\leqslant q^k|x_0-x^*|.\end{aligned} \tag{2-13}$$

又因为 $q<1$，当 $k\to\infty$ 时，由式(2-13)得
$$|x_k-x^*|\to 0,\quad 即\lim_{k\to\infty}x_k=x^*,$$
即是说，$\{x_k\}$ 对 $[a,b]$ 上的任意初值均收敛于 x^*.

再证明误差估计式(2-11).

由
$$\begin{aligned}|x_{k+1}-x_k|&\leqslant q|x_k-x_{k+1}|\\&\leqslant q^2|x_{k-1}-x_{k-2}|\\&\;\;\vdots\\&\leqslant q^k|x_1-x_0|,\end{aligned} \tag{2-14}$$

因为
$$x_{k+p}-x_k=(x_{k+p}-x_{k+p-1})+(x_{k+p-1}-x_{x+p-2})+\cdots+(x_{k+1}-x_k),$$
所以
$$|x_{k+p}-x_k|\leqslant|x_{k+p}-x_{k+p-1}|+|x_{k+p-1}-x_{x+p-2}|+\cdots+|x_{k+2}-x_{k+1}|+|x_{k+1}-x_k|.$$

据式(2-14)有
$$\begin{aligned}|x_{k+p}-x_k|&\leqslant q^{p-1}|x_{k+1}-x_k|+q^{p-2}|x_{k+1}-x_k|+\cdots+q|x_{k+1}-x_k|+|x_{k+1}-x_k|\\&=(q^{p-1}+q^{p-2}+\cdots q+1)|x_{k+1}-x_k|\\&=\frac{1-q^p}{1-q}|x_{k+1}-x_k|.\end{aligned} \tag{2-15}$$

固定 k，令 $p\to\infty$，由式(2-15)得

$$|x^* - x_k| \leqslant \frac{1}{1-q}|x_{k+1} - x_k|.$$

注 (1)上述不等式中 q 的指数规律为

$$(k+1)-1=k-0=k(\text{见式 }2-14).$$

$$\left.\begin{array}{l}(k+p)-(k+1)=(k+p-1)-k=p-1\\(k+p-1)-(k+1)=(k+p-2)-k=p-2\\\cdots\cdots\\(k+2)-(k+1)=(k+1)-k=1\end{array}\right\}(\text{见式 }2-15).$$

(2)误差估计式(2-11)的作用：

由 $q<1$，有 $0<1-q<1$. 当 $|x_{k+1}-x_k|<(1-q)\varepsilon=\varepsilon'$ 时，由式(2-11)可得 $|x_k-x^*|<\varepsilon$. 这就是迭代误差事后估计方法的理论依据.

定理 2.2 的条件很难检验，故在应用中通常采用局部收敛性定理.

定理 2.3(局部收敛判别定理) 设 $x_{k+1}=\varphi(x_k)$，$R=\{x\,|\,|x-x^*|<\delta\}$，若 $\varphi(x)$ 满足：

(1)$\varphi'(x)$ 在 R 内连续，

(2)$|\varphi'(x^*)|\leqslant q<1$，

则有与定理 2.2 相似的结论：

(1)$x_{k+1}=\varphi(x_k)$ 在 R 上惟一地收敛于 x^*，

(2)$|x_k-x^*|\leqslant \frac{1}{1-q}|x_{k+1}-x_k|$.

读者不难证明，当迭代函数 $\varphi(x)$ 满足条件(1)，(2)时，$\varphi(x)$ 在 R 上满足闭合条件和压缩条件，由此得到与定理 2.2 相似的结论. 但是定理 2.3 中的条件(2)不易检验，通常用 $|\varphi'(x_0)|\leqslant q\leqslant 1$ 替代，其中 $x_0\in R$.

例 11 设方程 $x^3-3x+1=0$，考察求解该方程的简单迭代式(2-2)，(2-3)及(2-4)在三个根附近的收敛性.

解 分别取三个存根区间内的初值为 $x_1=0.347$，$x_2=1.53$，$x_3=-1.88$.

(1)考察迭代公式(2-2)有

$$x_{k+1}=\frac{x_k^3+1}{3},\quad \varphi(x)=\frac{x^3+1}{3},\quad \varphi'(x)=x^2.$$

因为

$$|\varphi'(x_1)|=0.120\,409<1,\quad |\varphi'(x_2)|=2.340\,9>1,\quad |\varphi'(x_3)|=3.534\,4>1,$$

故据定理 2.3，迭代公式(2-2)在 x_1 附近收敛，而在 x_2，x_3 的附近判别失效.

(2)考察迭代公式(2-3)有

$$x_{k+1}=\frac{1}{3-x_k^2},\quad \varphi(x)=\frac{1}{3-x^2},\quad \varphi'(x)=\frac{2x}{(3-x^2)^2}.$$

因为

$$|\varphi'(x_1)|=0.08<1,\quad |\varphi'(x_2)|=7.04>1,\quad |\varphi'(x_3)|=13.16>1,$$

故据定理 2.3，迭代公式(2-3)在 x_1 附近收敛，而在 x_2，x_3 附近判别失效.

(3)考察迭代公式(2-4)有

$$x_{k+1}=\sqrt[3]{3x_k-1},\quad \varphi(x)=(3x-1)^{\frac{1}{3}},\quad \varphi'(x)=(3x-1)^{-\frac{2}{3}}.$$

因为

$$|\varphi'(x_1)|=1.135>1,\quad |\varphi'(x_2)|=0.744<1,\quad |\varphi'(x_3)|=0.395<1,$$

故据定理 2.3,迭代公式(2－4)在 x_2 及 x_3 附近收敛,而在 x_1 附近判别失效.

下面给出定理 2.3 的一个推论.

推论 2.1 Newton 迭代法在单实根邻近处局部收敛.

证 设 $f(x)=0$ 的一个单实根为 x^*,在 Newton 迭代公式 $x_{k+1}=x_k-\dfrac{f(x_k)}{f'(x_k)}$中

$$\varphi(x)=x-\frac{f(x)}{f'(x)},$$

$$\varphi'(x)=1-\frac{[f'(x)]^2-f(x)f''(x)}{[f'(x)]^2}=\frac{f(x)f''(x)}{[f'(x)]^2},$$

$$\varphi'(x^*)=\frac{f(x^*)f''(x^*)}{[f'(x^*)]^2}.$$

因为 x^* 为 $f(x)=0$ 的一个单实根,所以

$$f(x)=(x-x^*)\psi(x),\quad \text{且 } \psi(x^*)\neq 0.$$

而
$$f'(x^*)=(x^*-x^*)\psi'(x^*)+\psi(x^*)=\psi(x^*)\neq 0,$$

故
$$\varphi'(x^*)=0.$$

据定理 2.3,推论 2.1 成立.

例 12 方程 $xe^x-1=0$ 在[0,1]内有一个实根,考察迭代法的局部收敛性.

(1)考察简单迭代法.

由原方程构造简单迭代式有

$$x_{k+1}=e^{-x_k},$$

其中
$$\varphi(x)=e^{-x},\quad \varphi'(x)=-e^{-x}.$$

取 $x_0=0.5$,即有

$$\varphi'(x_0)=-e^{-0.5}=-0.606\,5,$$

$$|\varphi'(x_0)|\approx 0.6<1.$$

故 $x_{k+1}=e^{-x_k}$ 在 0.5 附近收敛于原方程的根.

(2)考察 Newton 迭代法.

由原方程构造 Newton 迭代式有

$$x_{k+1}=x_k-\frac{x_k-e^{-x_k}}{1+x_k}.$$

据推论 2.1,该迭代式在单根 $x^*\in[0,1]$附近局部收敛.

还可以证明上述简单迭代式及 Newton 迭代式的非局部收敛性,即在区间[0,2]上收敛于原方程的一个实根.

例 13 求$\sqrt{c}$ $(c\geqslant 0)$的近似值.

解 令 $x=\sqrt{c}$,得二次方程

$$x^2-c=0.$$

写出 Newton 迭代式

$$x_{k+1}=\frac{1}{2}\left(x_k+\frac{c}{x_k}\right).$$

据推论 2.1，当 x_0 在$\sqrt{c}$附近取值时，上述迭代式收敛于单根$\sqrt{c}$.

还可以证明，当 $x_0\in(0,+\infty)$时，该迭代式仍收敛于$\sqrt{c}$. 例如用迭代式 $x_{k+1}=\frac{1}{2}\left(x_k+\frac{115}{x_k}\right)$计算$\sqrt{115}$，取 $x_0\in(0,+\infty)$，该迭代式收敛.

例 14 求 $x-\tan x=0$ 的最小正根.

解 $y=\tan x$ 是以 π 为周期的周期函数，它与 $y=x$ 有无穷多个交点（即原方程的根）. 例如：在$\left(-\frac{\pi}{2},\frac{\pi}{2}\right)$内，它有一个实根 $x^*=0$；在$\left(-\frac{\pi}{2}+k\pi,\frac{\pi}{2}+k\pi\right)(k=1,2,\cdots)$内，它有无穷多个正根；在$\left(-\frac{\pi}{2}-k\pi,\frac{\pi}{2}-k\pi\right)(k=1,2,\cdots)$内，它有无穷多个负根；在$\left(\frac{\pi}{2},\frac{3}{2}\pi\right)$内，它有一个最小的正实根（如图 2－10 所示）.

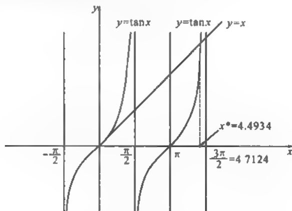


图 2－10

若用简单迭代法 $x_{k+1}=\tan x_k$，其中迭代函数为 $\varphi(x)=\tan x$. 因为

$$\varphi'(x)=\sec^2 x=1+\tan^2 x\geqslant 1,$$

所以收敛性判别失效.

现考虑 $x=\tan x$ 的反函数

$$x=\psi(x)=\arctan x.$$

因为 $\psi'(x)=\frac{1}{1+x^2}<1$，从而构成收敛的迭代式

$$x_{k+1}=\arctan x_k,$$

但由于 $x=\tan x$ 的最小正实根在$\left(\frac{\pi}{2},\frac{3}{2}\pi\right)$内，故只能采用迭代式

$$x_{k+1}=\pi+\arctan x_k.$$

取初值 $x_0=\pi+\frac{\pi}{6}=3.6652$，则

$x_1=4.4460$, $x_2=4.4911$, $x_3=4.4933$, $x_4=4.4934$, $x_5=4.4934$.
故取 $x^*\approx x_5=4.4934$(准确至第 4 位小数).

2.4.3 收敛阶(速度)及其判定

定义 2.4 设由 $x_{k+1}=\varphi(x_k)$所产生的点列$\{x_k\}$满足:

(1)$x_k\to x^*(k\to\infty)$,

(2)$e_k=x_k-x^*\neq 0, k=1,2,\cdots$

(3)$\lim\limits_{k\to\mu}\frac{|e_{k+1}|}{|e_k|^p}=c(c>0,p\geqslant 1)$,或$\exists K>0$,当 $k>K$ 时,$\frac{|e_{k+1}|}{|e_k|^p}\leqslant c$,

则称 $x_{k+1}=\varphi(x_k)$是 p 阶收敛的.

通常我们关注以下两种收敛速度:

(1)线性收敛($p=1$),即

$$\lim_{k\to\infty}\frac{|e_{k+1}|}{|e_k|}=c,\quad c\in(0,1),$$

或当 $k\geqslant K$ 时
$$|e_{k+1}|\leqslant c|e_k|. \tag{2-16}$$
据式(2-16)反复递推得$|e_k|\leqslant c^k|e_0|$,即
$$|x_k-x^*|\leqslant c^k|x_0-x^*|,\quad c<1. \tag{2-17}$$

由式(2-17)验证二分法为线性收敛.

设已进行了第 k 次二分,$x^*\in[a_k,b_k]$(如图 2-11).



图 2-11

因为
$$|x_k-x^*|\leqslant\frac{1}{2}|b_k-a_k|=\frac{1}{2}\frac{b-a}{2^k}\approx\frac{1}{2^k}|x_0-x^*|,$$
即
$$|x_k-x^*|\leqslant(\frac{1}{2})^k|x_0-x^*|,$$
由式(2-17)知二分法为线性收敛(此处 $c=\frac{1}{2}$).

(2)超线性收敛($p>1$),即

$$\lim_{k\to\infty}\frac{|e_{k+1}|}{|e_k|^p}=c>0,\quad p>1.$$

例如 Newton 法($p=2$),割线法($p=1.618$)均为超线性收敛.

按照定义 2.4 判别收敛阶相当困难,通常采用以下的充分性判别定理.

定理 2.4 设 $x_{k+1}=\varphi(x_k)$,$x^*=\varphi(x^*)$. 若 $\varphi(x)$满足:

(1)$\varphi^{(p)}(x)$在 x^* 邻近连续,

(2)$\varphi^{(k)}(x^*)=0, k=1,2,\cdots,p-1$,

(3)$\varphi^{(p)}=(x^*)\neq 0$, $p\geqslant 1$ 为正整数,

则 $x_{k+1}=\varphi(x_k)$在 x^* 邻近处 p 阶收敛,且有

$$\lim_{k\to\infty}\frac{e_{k+1}}{e_k^p}=\frac{1}{p!}\varphi^{(p)}(x^*).$$

证 $\varphi(x)=\varphi(x^*+(x-x^*))$

$$=\varphi(x^*)+\varphi'(x^*)(x-x^*)+\cdots+\frac{\varphi^{(p)}(\zeta)}{p!}(x-x^*)^p$$

$$=\varphi(x^*)+\frac{\varphi^{(p)}(\zeta)}{p!}(x-x^*)^p. \tag{2-18}$$

将 x_k 代入式(2－18)得

$$\varphi(x_k)=\varphi(x^*)+\frac{\varphi^{(p)}(\zeta)}{p!}(x_k-x^*)^p,$$

即

$$\frac{x_{k+1}-x^*}{(x_k-x^*)^p}=\frac{1}{p!}\varphi^{(p)}(\zeta).$$

其中

$$\zeta=x^*+\theta(x_k-x^*),\quad \theta\in(0,1),$$

$$\lim_{k\to\infty}\frac{|e^{k+1}|}{|e_k^p|}=\frac{1}{p!}\lim_{k\to\infty}|\varphi^{(p)}(\zeta)|=\frac{1}{p!}|\varphi^{(p)}(x^*)|\neq 0.$$

据定义 2.4 知，$x_{k+1}=\varphi(x_k)$在 x^* 邻近处 p 阶收敛.

推论 2.2 Newton 迭代法在单根 x^* 邻近处至少平方收敛.

证 在 Newton 迭代式中，迭代函数 $\varphi(x)=x-\dfrac{f(x)}{f'(x)}$，由推论 2.1 的证明知

$$\varphi'(x^*)=0,$$

而 $\varphi''(x^*)$只有以下两种情形：

(1)若 $\varphi''(x^*)\neq 0$，则由定理 2.4 知 $x_{k+1}=\varphi(x_k)$平方收敛；

(2)若 $\varphi''(x^*)=0$，则 $x_{k+1}=\varphi(x_k)$至少三阶收敛.

综合上述两种情形，Newton 迭代法在单根邻近处至少平方收敛.

例 15 已知方程 $x^3-x-1=0$ 在区间[1,2]内有一实根 $x^*=1.325$，讨论用简单迭代法求此实根的收敛速度问题.

解 (1)原方程的第一种等价形式为

$$x=(x+1)^{\frac{1}{3}},$$

形成简单迭代式为

$$x_{k+1}=(x_k+1)^{\frac{1}{3}}. \tag{2-19}$$

其中

$$\varphi(x)=(x+1)^{\frac{1}{3}},\quad \varphi'(x)=\frac{1}{3(x+1)^{\frac{2}{3}}}.$$

故 $\varphi'(1.325)=0.189\neq 0$. 据定理 2.4，迭代式(2－19)一阶收敛. 又因为

$$\lim_{k\to\infty}\frac{|e_{k+1}|}{|e_k|}=|\varphi'(1.325)|<1,$$

另据定义 2.4，迭代式(2－19)为线性收敛.

(2)原方程的第二种等价形式为

$$x=\frac{2x^3+1}{3x^2-1},$$

形成简单迭代式为

$$x_{k+1}=\frac{2x_k^3+1}{3x_k^2-1}. \tag{2-20}$$

其中 $$\varphi(x)=\frac{2x^3+1}{3x^2-1},\quad \varphi'(x)=\frac{6x(x^2-x-1)}{(3x^2-1)^2}.$$

故 $\varphi'(x^*)=0$. 据定理 2.4,迭代式(2-20)至少平方收敛.

事实上,简单迭代式(2-20)恰为 Newton 迭代式(见本章例 17). 因此,据推论 2.2,它也应具有至少平方收敛的结论. 该例说明,对于特定的方程 $f(x)=0$,通过特殊变形得到的简单迭代式可以具有与 Newton 迭代式相同的结构.

2.5 Newton 迭代法的应用

2.5.1 求重根和复根

2.5.1.1 求重根

设 $f(x)=0$,且

$$f(x)=(x-x^*)^m p(x),\quad p(x^*)\neq 0,\quad m>1.$$

令 $$F(x)=\sqrt[m]{f(x)}=(x-x^*)\cdot[p(x)]^{\frac{1}{m}},$$

即当 $f(x)$具有 m 重根时,$F(x)$具有单实根.

对 $F(x)=0$ 应用 Newton 迭代法有

$$x_{k+1}=x_k-\frac{F(x_k)}{F'(x_k)}=x_k-m\frac{f(x_k)}{f'(x_k)},$$

由此得到求 $f(x)=0$ 的 m 重根的 Newton 迭代公式

$$x_{k+1}=x_k-m\frac{f(x_k)}{f'(x_k)},\quad m>1. \tag{2-21}$$

例 16 求 $x^2+2xe^x+e^{2x}=0$ 在区间$[-1,0]$内的二重根.

解 设 $f(x)=x^2+2xe^x+e^{2x}=(x+e^x)^2$,据式(2-21)

$$x_{k+1}=x_k-2\frac{f(x_k)}{f'(x_k)}=\frac{e^{x_k}(x_k-1)}{1+e^{x_k}}.$$

取 $x_0=-1$,列表计算(保留 7 位小数),其结果见表 2-8.

表 2-8

k	x_k	k	x_k
0	−1	3	−0.567 143 3
1	−0.537 882 8	4	−0.567 143 3
2	−0.566 987 0	—	—

由表 2-8 可得 $x^*\approx x_4=-0.567\,143\,3$(准确至第 7 位小数).

若对原方程应用求单根的 Newton 迭代公式

$$x_{k+1}=x_k-\frac{f(x_k)}{f'(x_k)}=\frac{x_k+e^{x_k}(2x_k-1)}{2(1+e^{x_k})},$$

则收敛于重根的速度很慢.

2.5.1.2　求复根

Newton 迭代公式 $x_{k+1}=x_k-\frac{f(x_k)}{f'(x_k)}$ 不仅可以求单实根，也可以用来求复根，但初值必须取复数.

例 17　求 $f(x)=x^3-x-1=0$ 在区间[1,2]内的全部根.

方程 $f(x)=x^3-x-1=0$ 在区间[1,2]内除有一实根外还有一对复根(如图 2－12).

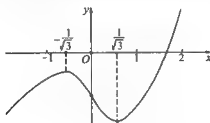


图 2－12

由 Newton 迭代式

$$x_{k+1}=x_k-\frac{x_k^3-x_k-1}{3x_k^2-1}=\frac{2x_k^3+1}{3x_k^2-1},$$

取 $x_0=1$，列表计算(保留 3 位小数)，其结果见表 2－9.

表 2－9

k	x_k	k	x_k
0	1	3	1.325
1	1.5	4	1.325
2	1.348	—	—

由表 2－9 可得 $x^*\approx x_4=1.325$(准确至第 3 位小数).

另取初值 $x_0=-0.5+0.5\mathrm{i}$，仍用上述 Newton 迭代式列表计算，其计算结果见表 2－10.

表 2－10

k	x_k	$f(x_k)$	$f'(x_k)$
0	$-0.5+0.5\mathrm{i}$	$-0.250-0.250\mathrm{i}$	$-1.000-1.500\mathrm{i}$
1	$0.6923+0.5385\mathrm{i}$	$-0.037+0.0795\mathrm{i}$	$-0.432-2.237\mathrm{i}$
2	$-0.6611+0.5611\mathrm{i}$	$-0.002-0.0022\mathrm{i}$	$-0.632-2.225\mathrm{i}$
3	$-0.6624+0.5627\mathrm{i}$	—	—

由表 2－10 可得原方程的一对复根：$x_{2,3}=-0.6624\pm0.5627\mathrm{i}$.

2.5.2 Newton下降法

如前所述，Newton迭代法在单根附近具有较快的收敛速度（至少平方收敛）。因此在运用该方法时初始点 x_0 应取在单根 x^* 的附近。当 x_0 偏离 x^* 较远时，在某些情形下Newton迭代法甚至不能保证收敛。

在2.5.1的例17中，三次方程 $x^3-x-1=0$ 在区间[1,2]内有一个实根（如图2-13）。若初始点 x_0 在区间[1,2]内选取，用Newton迭代公式会很快地收敛于此实根；若初始点 $x_0\leqslant\frac{1}{\sqrt{3}}$，则Newton迭代公式不收敛。

为了防止因初始点 x_0 远离 x^* 导致迭代不收敛，通常采用Newton下降法（或阻尼方法）。

Newton下降法的迭代公式为

$$\begin{cases}\bar{x}_{k+1}=x_k-\dfrac{f(x_k)}{f'(x_k)},\\ x_{k+1}=\lambda\bar{x}_{k+1}+(1-\lambda)x_k.\end{cases}\tag{2-22}$$

式中，$\lambda\in(0,1]$ 为阻尼因子，λ 应满足 $|f(x_{k+1})|<|f(x_k)|$。

Algo. 2.5（Newton下降法）

1° 送初值 x_0 及预给精度 $\varepsilon_1,\varepsilon_2>0$。

2° 计算 $x_1=x_0-\dfrac{f(x_0)}{f'(x_0)}$。

3° 令 $\lambda=1$。

4° 计算 $x_2=\lambda x_1+(1-\lambda)x_0$。

5° 检查：$|f(x_2)|<|f(x_0)|$？

是，转7°（当前迭代成功，转入下一步迭代）；

否，转6°（缩小 λ 值，继续当前迭代）。

6° 检查：$|\lambda|<\varepsilon_1$？

是，stop（迭代失败的出口）；

否，$\dfrac{\lambda}{2}\Rightarrow\lambda$，转4°。

7° $x_2\Rightarrow x_1$。

8° 检查：$|x_1-x_0|<\varepsilon_2$ 或 $\dfrac{|x_1-x_0|}{|x_1|}<\varepsilon_2$？

是，输出 x_1，　stop；

否，$x_1\Rightarrow x_0$，转2°。

习题2

*1. 用二分法求下列方程的根，要求绝对误差限为0.005。

(1) $x^3-x+4=0$，求根区间为(-2,-1)。

(2) $2x=e^{-x}$，求根区间为(0,1)。

*2. 用二分法搜索方程 $6x^3-4x-1=0$ 的实根分布情况，初始搜索区间为$(-2,2)$，并求出$(0,1)$中的根，精确到 0.05.

*3. 验证本章例 6 中迭代公式

$$x_{k+1}=\frac{x_k^3+1}{3}$$

不收敛于 1.5 和 -2 附近的根.

*4. 用迭代法求 $x^3-2x-5=0$ 的正根. 下面有三种迭代公式：

$$(1)x=\frac{x^3-5}{2},\quad (2)x=\frac{5}{x^2-2},\quad (3)x=\sqrt[3]{2x+5}.$$

先判定它们在 $x_0=2$ 附近的收敛情况，并选择收敛的方法求出此根，要求结果精确到 4 位小数.

*5. 用松驰法将第 4 题迭代公式(3)加速，并求 $x_0=2$ 附近的根，使其精确到 4 位小数.

*6. 用迭代法计算 $x=\cos x$ 的正根收敛较慢. 试用 Aitken 方法加速收敛，求此根，使其精确到 4 位小数，并与迭代法的收敛速度比较.

*7. 用 Newton 迭代法：

(1)不开平方计算 $\sqrt{10}$，且使其结果精确到 5 位小数；

(2)计算方程 $x^3-x-1=0$ 在 1.3 附近的根，精确到 5 位小数；

(3)计算 $2x=e^{-x}$ 的根，精确到 4 位小数，并与 1 题中(2)的收敛速度作比较.

8. 已知 $x=\Phi(x)$ 在区间$[a,b]$内只有一根，而当 $a<x<b$ 时，

$$|\Phi'(x)|\geqslant k>1.$$

试问如何将 $x=\Phi(x)$ 化为适于迭代的形式?

将 $x=\tan x$ 化为适于迭代的形式，并求 $x=4.5$(弧度)附近的根.

9. 研究求$\sqrt{a}$的 Newton 迭代公式

$$x_{k+1}=\frac{1}{2}\left(x_k+\frac{a}{x_k}\right),$$

证明：对一切 $k=1,2,\cdots$来说，$x_k>\sqrt{a}$，且序列 $x_1,x_2,\cdots$是严格递减的.

10. 应用 Newton 迭代法于方程 $f(x)=x^n-a$ 和 $f(x)=1-\frac{a}{x^n}$，分别导出求$\sqrt[n]{a}$的迭代公式，并求

$$\lim_{k\to\infty}\frac{(\sqrt[n]{a}-x_{k+1})}{(\sqrt[n]{a}-x_k)^2}.$$

11. 证明迭代公式

$$x_{k+1}=\frac{x_k(x_k^2+3a)}{3x_k^2+a}$$

是计算$\sqrt{a}$的三阶方法. 假定初值 x_0 充分靠近根 x^*，求$\lim\limits_{k\to\infty}\frac{(\sqrt{a}-x_{k+1})}{(\sqrt{a}-x_k)^3}$.

注　打 * 号的习题在本书附录 II 作了详解.

第3章 线性方程组的直接解法

解线性方程组是科学研究与工程计算中经常遇到的问题.在有些问题的数学模型中虽然不直接含有线性方程组,但在它的数值解法中常常须将问题“离散化”或“线性化”为线性方程组.因此求解线性方程组是本课程的基本内容.

3.1 引言

3.1.1 线性方程组的分类

含 n 个未知数 $x_i(i=1,2,\cdots,n)$ 的 m 个线性等式构成线性方程组

$$\sum_{j=1}^{n} a_{ij}x_j = b_i, \quad i=1,2,\cdots,m. \tag{3-1}$$

其中,系数 $a_{ij}(i=1,2,\cdots,m;j=1,2,\cdots,n)$ 及右端 $b_i(i=1,2,\cdots,m)$ 均为已知数,而 $x_j(j=1,2,\cdots,n)$ 为未知数.

根据 m 与 n 的大小关系,线性方程组(3-1)可分为以下3种类型:

(1)$m=n$,称为适定方程组;

(2)$m<n$,称为不定方程组;

(3)$m>n$,称为超定方程组.

不定(或超定)方程组均须化为适定方程组才能求解.

3.1.2 线性方程组的矩阵形式

将方程组(3-1)中的 m 个线性等式写成

$$(a_{11}a_{12}\cdots a_{1n})\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}=b_1,$$

$$(a_{21}a_{22}\cdots a_{2n})\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}=b_2,$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$(a_{m1}a_{m2}\cdots a_{mn})\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}=b_m.$$

然后合并上式得到矩阵形式

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}.$$

令

$$\boldsymbol{A}=(a_{ij})_{m\times n}, \quad \boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}, \quad \boldsymbol{b}=(b_1,b_2,\cdots,b_m)^{\mathrm{T}},$$

则有

$$\boldsymbol{Ax}=\boldsymbol{b}. \tag{3-2}$$

称式(3-2)为线性方程组的矩阵形式.

本章讨论适定方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 的解法,其中

$$\boldsymbol{A}=(a_{ij})_{n\times n}, \quad \boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}, \quad \boldsymbol{b}=(b_1,b_2,\cdots,b_n)^{\mathrm{T}}.$$

3.1.3 线性方程组解的存在惟一性

设有适定方程组

$$\boldsymbol{Ax}=\boldsymbol{b}.$$

(1)若 $\boldsymbol{b}\neq\boldsymbol{0}$,则 $\boldsymbol{Ax}=\boldsymbol{b}$ 存在惟一解的充分必要条件是

$$\det(\boldsymbol{A})\neq 0;$$

(2)若 $\boldsymbol{b}=\boldsymbol{0}$,则 $\boldsymbol{Ax}=\boldsymbol{b}$ 必有平凡解 $\boldsymbol{x}=\boldsymbol{0}$. 该方程组存在非平凡解的充分必要条件是

$$\det(\boldsymbol{A})=0.$$

3.1.4 线性方程组的解法

3.1.4.1 公式法(Gramer 法则)

设方程组 $\boldsymbol{Ax}=\boldsymbol{b},\boldsymbol{b}\neq\boldsymbol{0},\det(\boldsymbol{A})\neq 0$,则方程组存在惟一解

$$x_j=\Delta_j/\det(\boldsymbol{A}), \quad j=1,2,\cdots,n.$$

其中 $$\det(\boldsymbol{A})=\begin{vmatrix} a_{11} & \cdots & a_{1j} & \cdots & a_{1n} \\ a_{21} & \cdots & a_{2j} & \cdots & a_{2n} \\ \vdots & & \vdots & & \vdots \\ a_{n1} & \cdots & a_{nj} & \cdots & a_{nn} \end{vmatrix}, \quad \Delta_j=\begin{vmatrix} a_{11} & \cdots & b_1 & \cdots & a_{1n} \\ a_{21} & \cdots & b_2 & \cdots & a_{2n} \\ \vdots & & \vdots & & \vdots \\ a_{n1} & \cdots & b_n & \cdots & a_{nn} \end{vmatrix}.$$

求解 $\boldsymbol{Ax}=\boldsymbol{b}$ 的计算量为计算 $n+1$ 个行列式所需乘法的次数再加上 n 次除法的次数,即为

$$(n+1)!+n.$$

如设 $n=10$,则需要的乘除法次数为 39 916 810. 若用计算速度为每秒 10 万次的计算机进行计算,则所需时间为 400 s.

3.1.4.2 高斯(Gauss)消元法(直接解法)

Gauss 消元法消元步和回代步需用乘除法的次数共计为 $\frac{n^3}{3}+n^2-\frac{n}{3}$,当 n 较大时,约为 $\frac{n^3}{3}$. 设 $n=10$,则需用乘除法的次数共计为 430 次. 若用计算速度为每秒 10 万次的计算

机进行计算,所需时间不到$\frac{5}{1\,000}$ s.

由此可见,利用计算机求解同样的方程组,Gauss 消元法要比公式法提高效率80 000倍.

3.2 高斯(Gauss)消元法

Gauss 消元法是一个古老的直接法,由它改进得到的主元素消去法,是目前计算机上用于求解低阶稠密矩阵方程组的有效方法。

3.2.1 Gauss 顺序消元法

设
$$Ax=b,$$
其中 $A=(a_{ij})_{n\times n}, b=(b_1,b_2,\cdots,b_n)^{\mathrm{T}}$. 令

$$A=A^{(1)}=(a_{ij}^{(1)})_{n\times n},\quad \det A^{(1)}\neq 0,$$
$$b=b^{(1)}=(b_1^{(1)},b_2^{(1)},\cdots,b_n^{(1)})^{\mathrm{T}},\quad b^{(1)}\neq 0.$$

用 Gauss 顺序消元法解 $Ax=b$,分为如下两个步骤.

3.2.1.1 消元步

Gauss 消元法的消元步是完成如下增广矩阵的变换,即

$$(A^{(1)}\mid b^{(1)})\xrightarrow[\text{消元}]{n-1\text{ 次}}(A^{(n)}\mid b^{(n)}).$$

其中

$$A^{(n)}=\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ & & \ddots & \vdots \\ & & & a_{nn}^{(n)} \end{bmatrix},\quad b^{(n)}=(b_1^{(1)},b_2^{(2)},\cdots,b_n^{(n)})^{\mathrm{T}}.$$

原方程组 $Ax=b$,即 $A^{(1)}x=b^{(1)}$ 与三角方程组 $A^{(n)}x=b^{(n)}$ 等价. 也就是说,$A^{(n)}x=b^{(n)}$ 的解就是 $A^{(1)}x=b^{(1)}$ 的解,亦即 $Ax=b$ 的解. 下面分析如何由$(A^{(1)}\mid b^{(1)})$转化为$(A^{(n)}\mid b^{(n)})$.

第一次消元 设 $a_{11}^{(1)}\neq 0$.

$$(A^{(1)}\mid b^{(1)})\xrightarrow{\text{初等变换}}(A^{(2)}\mid b^{(2)}),$$

即

$$\left[\begin{array}{cccccc:c} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1j}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & \cdots & a_{2j}^{(1)} & \cdots & a_{2n}^{(1)} & b_2^{(1)} \\ \vdots & \vdots & & \vdots & & \vdots & \vdots \\ a_{i1}^{(1)} & a_{i2}^{(1)} & \cdots & a_{ij}^{(1)} & \cdots & a_{in}^{(1)} & b_i^{(1)} \\ \vdots & \vdots & & \vdots & & \vdots & \vdots \\ a_{n1}^{(1)} & a_{n2}^{(1)} & \cdots & a_{nj}^{(1)} & \cdots & a_{nn}^{(1)} & b_n^{(1)} \end{array}\right] \begin{array}{l} \\ m_{21}=a_{21}^{(1)}/a_{11}^{(1)} \\ \vdots \\ m_{i1}=a_{i1}^{(1)}/a_{11}^{(1)} \\ \vdots \\ m_{n1}=a_{n1}^{(1)}/a_{11}^{(1)} \end{array}$$

$$\downarrow$$

$$\left[\begin{array}{cccccc:c} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1j}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2j}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\ \vdots & \vdots & & \vdots & & \vdots & \vdots \\ 0 & a_{i2}^{(2)} & \cdots & a_{ij}^{(2)} & \cdots & a_{in}^{(2)} & b_i^{(2)} \\ \vdots & \vdots & & \vdots & & \vdots & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nj}^{(2)} & \cdots & a_{nn}^{(2)} & b_n^{(2)} \end{array}\right].$$

第一次消元执行下列计算：

$$\begin{cases} m_{i1}=a_{i1}^{(1)}/a_{11}^{(1)}, & i=2,3,\cdots,n; \\ a_{ij}^{(2)}=a_{ij}^{(1)}-m_{i1}a_{1j}^{(1)}, & i,j=2,3,\cdots,n; \\ b_i^{(2)}=b_i^{(1)}-m_{i1}b_1^{(1)}, & i=2,3,\cdots,n. \end{cases} \tag{3-3}$$

式中，$m_{i1}(i=2,3,\cdots,n)$称为辅助乘数，引人该乘数是为了计算$(A^{(2)}\mid b^{(2)})$中除去第一行的其他各行元素，即

$$a_{ij}^{(2)},\quad i,j=2,3,\cdots,n;\qquad b_i^{(2)},\quad i=2,3,\cdots,n.$$

设已进行了 $k-1$ 次消元，得到 $A^{(k)}$ 和 $b^{(k)}$.

第 k 次消元 设 $a_{kk}^{(k)}\neq 0$.

$$(A^{(k)}\mid b^{(k)})\xrightarrow{\text{初等变换}}(A^{(k+1)}\mid b^{(k+1)}),$$

即

$$\left[\begin{array}{cccccc:c} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1k}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ & a_{22}^{(2)} & \cdots & a_{2k}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\ & & \ddots & \vdots & & \vdots & \vdots \\ & & & \ddots & & & \\ & & & a_{kk}^{(k)} & \cdots & a_{kn}^{k} & b_k^{(k)} \\ & & & a_{k+1,k}^{(k)} & \cdots & a_{k+1,n}^{(k)} & b_{k+1}^{(k)} \\ & & & \vdots & & & \vdots \\ & & & a_{nk}^{(k)} & \cdots & a_{nn}^{(k)} & b_n^{(k)} \end{array}\right] \begin{array}{l} \\ \\ \\ \\ \\ m_{k+1,k}=a_{k+1,k}^{(k)}/a_{kk}^{(k)} \\ \vdots \\ m_{nk}=a_{nk}^{(k)}/a_{kk}^{(k)} \end{array}$$

$$\downarrow$$

$$\left[\begin{array}{cccccccc:c} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1k}^{(1)} & a_{1,k+1}^{(1)} & \cdots & a_{1n}^{(1)} & & b_1^{(1)} \\ & a_{22}^{(2)} & \cdots & a_{2k}^{(2)} & a_{2,k+1}^{(2)} & \cdots & a_{2n}^{(2)} & & b_2^{(2)} \\ & & \ddots & \vdots & \vdots & & \vdots & & \vdots \\ & & & \ddots & & & & & \\ & & & a_{kk}^{(k)} & a_{k,k+1}^{(k)} & \cdots & a_{kn}^{k} & & b_k^{(k)} \\ & & & 0 & a_{k+1,k+1}^{(k+1)} & \cdots & a_{k+1,n}^{(k+1)} & & b_{k+1}^{(k+1)} \\ & & & \vdots & \vdots & & \vdots & & \vdots \\ & & & 0 & a_{n,k+1}^{(k+1)} & \cdots & a_{nn}^{(k+1)} & & b_n^{(k+1)} \end{array}\right].$$

第 k 次消元执行下列计算：

$$\begin{cases} m_{ik}=a_{ik}^{(k)}/a_{kk}^{(k)}, & i=k+1,\cdots,n; \\ a_{ij}^{(k+1)}=a_{ij}^{(k)}-m_{ik}a_{kj}^{(k)}, & i,j=k+1,\cdots,n; \\ b_i^{(k+1)}=b_i^{(k)}-m_{ik}b_k^{(k)}, & i=k+1,\cdots,n. \end{cases} \tag{3-4}$$

令 $k=1,2,\cdots,n-1$. 若执行上述三步计算，则完成 $n-1$ 次消元，即将增广矩阵 $(\boldsymbol{A}^{(1)}|\boldsymbol{b}^{(1)})$ 转化为 $(\boldsymbol{A}^{(n)}\ \ \boldsymbol{b}^{(n)})$，也即将解方程组 $\boldsymbol{A}^{(1)}\boldsymbol{x}=\boldsymbol{b}^{(1)}$ 转化为解三角方程组 $\boldsymbol{A}^{(n)}\boldsymbol{x}=\boldsymbol{b}^{(n)}$. 其中

$$\boldsymbol{A}^{(n)}=\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ & & \ddots & \vdots \\ & & & \ddots \\ & & & a_{nn}^{(n)} \end{bmatrix}, \quad \boldsymbol{b}^{(n)}=(b_1^{(1)},b_2^{(2)},\cdots,b_n^{(n)})^{\mathrm{T}}.$$

3.2.1.2 回代步

Gauss 消元法的回代步就是求解三角方程组 $\boldsymbol{A}^{(n)}\boldsymbol{x}=\boldsymbol{b}^{(n)}$，它的解 $\boldsymbol{x}$ 就是原方程组 $\boldsymbol{A}^{(1)}\boldsymbol{x}=\boldsymbol{b}^{(1)}$ 的解.

$\boldsymbol{A}^{(n)}\boldsymbol{x}=\boldsymbol{b}^{(n)}$ 的分量形式为

$$\sum_{j=i}^{n} a_{ij}^{(i)}x_j=b_i^{(i)}, \quad i=1,2,\cdots,n.$$

令 $i=n$，可解得 x_n；

令 $i=n-1$，可解得 x_{n-1}；

$\vdots$

令 $i=1$，可解得 x_1.

求解 x_n，由第 n 个方程得到，即

$$a_{nn}^{(n)}x_n=b_n^{(n)}, \quad x_n=b_n^{(n)}/a_{nn}^{(n)}.$$

求解 $x_i(i=n-1,n-2,\cdots,1)$ 的一般公式由第 i 个方程得到，即

$$a_{ii}^{(i)}x_i+\sum_{j=i+1}^{n}a_{ij}^{(i)}x_j=b_i^{(i)},$$

$$x_i=(b_i^{(i)}-\sum_{j=i+1}^{n}a_{ij}^{(i)}x_j)/a_{ii}^{(i)}, \quad i=n-1,n-2,\cdots,1.$$

于是我们得到回代公式

$$\begin{cases} x_n=b_n^{(n)}/a_{nn}^{(n)}, \\ x_i=(b_i^{(i)}-\sum_{j=i+1}^{n}a_{ij}^{(i)}x_j)/a_{ii}^{(i)}, \quad i=n-1,n-2,\cdots,1. \end{cases} \tag{3-5}$$

3.2.1.3 *Gauss 顺序消元法计算步骤*

1)消元步

令 $k=1,2,\cdots,n-1$，按式(3-4)计算：

$$\begin{cases} m_{ik}=a_{ik}^{(k)}/a_{kk}^{(k)}, & i=k+1,\cdots,n; \\ a_{ij}^{(k+1)}=a_{ij}^{(k)}-m_{ik}a_{kj}^{(k)}, & i,j=k+1,\cdots,n; \\ b_i^{(k+1)}=b_i^{(k)}-m_{ik}b_k^{(k)}, & i=k+1,\cdots,m. \end{cases}$$

2)回代步

回代过程按式(3-5)计算,即

$$\begin{cases} x_n = b_n^{(n)}/a_{nn}^{(n)}, \\ x_i = (b_i^{(i)} - \sum_{j=i+1}^{n} a_{ij}^{(i)} x_j)/a_{ii}^{(i)}, \quad i = n-1, n-2, \cdots, 1. \end{cases}$$

3.2.1.4 计算表格

以三阶方程组为例,列表求解.

$$\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \vdots & b_1^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & a_{23}^{(1)} & \vdots & b_2^{(1)} \\ a_{31}^{(1)} & a_{32}^{(1)} & a_{33}^{(1)} & \vdots & b_3^{(1)} \end{bmatrix} \begin{matrix} \\ m_{21} = a_{21}^{(1)}/a_{11}^{(1)} \\ m_{31} = a_{31}^{(1)}/a_{11}^{(1)} \end{matrix}$$

$$\downarrow (1)$$

$$\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \vdots & b_1^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \vdots & b_2^{(2)} \\ 0 & a_{32}^{(2)} & a_{33}^{(2)} & \vdots & b_3^{(2)} \end{bmatrix} m_{32} = a_{32}^{(2)}/a_{22}^{(2)}$$

$$\downarrow (2)$$

$$\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \vdots & b_1^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \vdots & b_2^{(2)} \\ 0 & 0 & a_{33}^{(3)} & \vdots & b_3^{(3)} \end{bmatrix}$$

即得

$$x_3 = b_3^{(3)}/a_{33}^{(3)},$$
$$x_2 = (b_2^{(2)} - a_{23}^{(2)} x_3)/a_{22}^{(2)},$$
$$x_1 = (b_1^{(1)} - a_{12}^{(1)} x_2 - a_{13}^{(1)} x_3)/a_{11}^{(1)}.$$

原方程组的解为 $\boldsymbol{x} = (x_1, x_2, x_3)^{\mathrm{T}}$.

例 1 用 Gauss 顺序消元法解方程组 $\boldsymbol{Ax} = \boldsymbol{b}$,其中

$$\boldsymbol{A} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 7 & 5 \\ 1 & 4 & 9 \end{bmatrix}, \quad \boldsymbol{b} = (1, 6, -3)^{\mathrm{T}}.$$

解

$$\begin{bmatrix} 1 & 2 & 3 & \vdots & 1 \\ 2 & 7 & 5 & \vdots & 6 \\ 1 & 4 & 9 & \vdots & -3 \end{bmatrix} \begin{matrix} \\ m_{21} = 2 \\ m_{31} = 1 \end{matrix}$$

$$\downarrow (1)$$

$$\begin{bmatrix} 1 & 2 & 3 & \vdots & 1 \\ 0 & 3 & -1 & \vdots & 4 \\ 0 & 2 & 6 & \vdots & -4 \end{bmatrix} m_{32} = \frac{2}{3}$$

$$\downarrow (2)$$

$$\begin{bmatrix} 1 & 2 & 3 & \vdots & 1 \\ 0 & 3 & -1 & \vdots & 4 \\ 0 & 0 & \frac{20}{3} & \vdots & -\frac{20}{3} \end{bmatrix},$$

即
$$x_3=-1,\quad x_2=1,\quad x_1=2.$$

所以,原方程组的解为 $\boldsymbol{x}=(2,1,-1)^{\mathrm{T}}$.

例 2 用 Gauss 顺序消元法解方程组 $\boldsymbol{Ax}=\boldsymbol{b}$,其中

$$\boldsymbol{A}=\begin{bmatrix} 3 & -2 & -2 \\ 1 & -\frac{2}{3} & -\frac{1}{3} \\ -1 & -4 & 1 \end{bmatrix},\quad \boldsymbol{b}=(1,0,-10)^{\mathrm{T}}.$$

解

$$\begin{bmatrix} 3 & -2 & -2 & \vdots & 1 \\ 1 & -\frac{2}{3} & -\frac{1}{3} & \vdots & 0 \\ -1 & -4 & 1 & \vdots & -10 \end{bmatrix} \begin{matrix} m_{21}=\frac{1}{3} \\ m_{31}=-\frac{1}{3} \end{matrix}$$

$$\downarrow (1)$$

$$\begin{bmatrix} 3 & -2 & -2 & \vdots & 1 \\ 0 & 0 & \frac{1}{3} & \vdots & -\frac{1}{3} \\ 0 & \boxed{-\frac{14}{3}} & \frac{1}{3} & \vdots & -\frac{29}{3} \end{bmatrix}$$

$$\downarrow r_2 \xleftrightarrow{\text{换行}} r_3$$

$$\begin{bmatrix} 3 & -2 & -2 & \vdots & 1 \\ 0 & -\frac{14}{3} & \frac{1}{3} & \vdots & -\frac{29}{3} \\ 0 & 0 & \frac{1}{3} & \vdots & -\frac{1}{3} \end{bmatrix}$$

即
$$x_3=-1,\quad x_2=2,\quad x_1=1.$$

所以,原方程的解为 $\boldsymbol{x}=(1,2,-1)^{\mathrm{T}}$.

3.2.2 Gauss 顺序消元法的条件

引理 3.1 设 $\boldsymbol{Ax}=\boldsymbol{b}$,并令 $\boldsymbol{A}^{(1)}=\boldsymbol{A},\boldsymbol{b}^{(1)}=\boldsymbol{b}$,则

$$(\boldsymbol{A}^{(1)}\mid\boldsymbol{b}^{(1)})\xrightarrow[\text{消元}]{n-1\text{次}}(\boldsymbol{A}^{(n)}\mid\boldsymbol{b}^{(n)})$$

的必要充分条件(以下简称必充条件)是

$$a_{ii}^{(i)}\neq 0,\quad i=1,2,\cdots,n-1.$$

证　第一次消元，即完成如下变换：

$$(A^{(1)}\mid b^{(1)})\longrightarrow(A^{(2)}\mid b^{(2)}).$$

计算 $m_{i1}=a_{i1}^{(1)}/a_{11}^{(1)}(i=2,3,\cdots,n)$ 的必充条件是 $a_{11}^{(1)}\neq0$.

第二次消元，即完成如下变换：

$$(A^{(2)}\mid b^{(2)})\longrightarrow(A^{(3)}\mid b^{(3)}).$$

计算 $m_{i2}=a_{i2}^{(2)}/a_{22}^{(2)}(i=3,4,\cdots,n)$ 的必充条件是 $a_{22}^{(2)}\neq0$.

第 $n-1$ 次消元，即完成如下变换：

$$(A^{(n-1)}\mid b^{(n-1)})\longrightarrow(A^{(n)}\mid b^{(n)}),$$

其中

$$(A^{(n-1)}\mid b^{(n-1)})=\left[\begin{array}{ccccc:c} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & & a_{1n}^{(1)} & b_1^{(1)} \\ & a_{22}^{(2)} & \cdots & & a_{2n}^{(2)} & b_2^{(2)} \\ & & \ddots & & \vdots & \vdots \\ & & & a_{n-1,n-1}^{(n-1)} & a_{n-1,n}^{(n-1)} & b_{n-1}^{(n-1)} \\ & & & a_{n,n-1}^{(n-1)} & a_{nn}^{(n-1)} & b_n^{(n-1)} \end{array}\right].$$

计算 $m_{n,n-1}=a_{n,n-1}^{(n-1)}/a_{n-1,n-1}^{(n-1)}$ 的必充条件是 $a_{n-1,n-1}^{(n-1)}\neq0$.

综上，执行 Gauss 顺序消元法的必充条件是

$$a_{ii}^{(i)}\neq0,\quad i=1,2,\cdots,n-1.$$

定理 3.1　设 $Ax=b$，并令 $A^{(1)}=A,b^{(1)}=b$，则

$$(A^{(1)}\mid b^{(1)})\xrightarrow[\text{消元}]{n-1\text{ 次}}(A^{(n)}\mid b^{(n)})$$

的必充条件是

$$\det(A_i)\neq0,\quad i=1,2,\cdots,n-1.$$

证　因为

$$\det(A_1)=a_{11}=a_{11}^{(1)},$$

$$\det(A_2)=\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}=\begin{vmatrix} a_{11}^{(1)} & a_{12}^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} \end{vmatrix}=\begin{vmatrix} a_{11}^{(1)} & a_{12}^{(1)} \\ 0 & a_{22}^{(2)} \end{vmatrix}=a_{11}^{(1)}\cdot a_{22}^{(2)},$$

$$\vdots$$

$$\det(A_i)=\begin{vmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1i}^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & \cdots & a_{2i}^{(1)} \\ \vdots & \vdots & & \vdots \\ a_{i1}^{(1)} & a_{i2}^{(1)} & \cdots & a_{ii}^{(1)} \end{vmatrix}=\begin{vmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1i}^{(1)} \\ & a_{22}^{(2)} & \cdots & a_{2i}^{(2)} \\ & & \ddots & \vdots \\ & & & a_{ii}^{(i)} \end{vmatrix}$$

$$=a_{11}^{(1)}a_{22}^{(2)}\cdots a_{ii}^{(i)}=\det(A_{i-1})\cdot a_{ii}^{(i)}.$$

综上得

$$\begin{cases}\det(A_1)=a_{11}^{(1)},\\ \det(A_i)=\det(A_{(i-1)})\cdot a_{ii}^{(i)},\quad i=2,3,\cdots,n.\end{cases}$$

所以，$\det(A_i)\neq0,i=1,2,\cdots,n-1$ 的必充条件是

$$a_{ii}^{(i)}\neq0,\quad i=1,2,\cdots,n-1.$$

根据引理 3.1，则

$$(A^{(1)}\mid b^{(1)})\xrightarrow[\text{消元}]{n-1\text{ 次}}(A^{(n)}\mid b^{(n)})$$

的必充条件是

$$\det(A_i)\neq 0,\quad i=1,2,\cdots,n-1.$$

注 引理 3.1 及定理 3.1 是讲消元步骤的必充条件，而 Gauss 消元法还有回代步骤，故执行 Gauss 顺序消元的必充条件应是 $a_{ii}^{(i)}\neq 0, i=1,2,\cdots,n$(回代时 $a_{nn}^{(n)}\neq 0$).

3.3 选主元的 Gauss 消元法

3.3.1 Gauss 列主元消元法

3.3.1.1 引例

例 3 解方程组 $Ax=b$，其中

$$A=\begin{pmatrix}0.000\,01 & 1\\ 2 & 1\end{pmatrix},\quad b=(1.000\,01,3)^{\mathrm{T}}.$$

计算中要求保留 4 位有效数字.

解 若用 Gauss 顺序消元法求解，有

$$\left[\begin{array}{cc:c}0.100\,0\times10^{-4} & 0.100\,0\times10 & 0.100\,0\times10\\ 0.200\,0\times10 & 0.100\,0\times10 & 0.300\,0\times10\end{array}\right] m_{21}=0.200\,0\times10^{6}\quad(\text{此值很大})$$

$$\downarrow$$

$$\left[\begin{array}{cc:c}0.100\,0\times10^{-4} & 0.100\,0\times10 & 0.100\,0\times10\\ 0 & -0.200\,0\times10^{6} & -0.200\,0\times10^{6}\end{array}\right],$$

即

$$x_2=1,\quad x_1=0.$$

由此得 Gauss 顺序消元法的解为 $\bar{x}=(0,1)^{\mathrm{T}}$，而原方程组的精确解为 $x^*=(1,1)^{\mathrm{T}}$.

$\bar{x}$的第一个分量之所以产生如此大的偏差，是因为 $a_{11}=0.100\,0\times10^{-4}$. 它的绝对值很小，以它为除数得到的 m_{21} 值很大，这就必然放大了原始数据的误差，导致解的失真.

为了解决上述问题，通常采用主元素消元法. 这里介绍 Gauss 列主元消元法(即按列选取主元素).

仍以例 3 为例，选取 0.200 0×10 作为主元，则有

$$\left[\begin{array}{cc:c}0.100\,0\times10^{-4} & 0.100\,0\times10 & 0.100\,0\times10\\ \boxed{0.200\,0\times10} & 0.100\,0\times10 & 0.300\,0\times10\end{array}\right]$$

$$\downarrow r_1\leftrightarrow r_2$$

$$\left[\begin{array}{cc:c}0.200\,0\times10 & 0.100\,0\times10 & 0.300\,0\times10\\ 0.100\,0\times10^{-4} & 0.100\,0\times10 & 0.100\,0\times10\end{array}\right] m_{21}=0.500\,0\times10^{-5}\quad(\text{此值很小})$$

$$\downarrow$$

$$\left[\begin{array}{cc:c}0.200\,0\times10 & 0.100\,0\times10 & 0.300\,0\times10\\ 0 & 0.100\,0\times10 & 0.100\,0\times10\end{array}\right],$$

即

$$x_2=1,\quad x_1=1.$$

由此得 Gauss 列主元消元法的解为 $x=(1,1)^{\mathrm{T}}$，它与精确解完全一致.

例 4 用 Gauss 列主元消元法解方程组 $\boldsymbol{Ax}=\boldsymbol{b}$,其中

$$\boldsymbol{A}=\begin{bmatrix}1&2&3\\2&4&5\\3&5&6\end{bmatrix},\quad \boldsymbol{b}=(1,0,0)^{\mathrm{T}}.$$

解

$$\left[\begin{array}{ccc:c}1&2&3&1\\2&4&5&0\\\boxed{3}&5&6&0\end{array}\right]\xrightarrow{r_1\leftrightarrow r_3}\left[\begin{array}{ccc:c}3&5&6&0\\2&4&5&0\\1&2&3&1\end{array}\right]\begin{matrix}\\m_{21}=2/3\\m_{31}=1/3\end{matrix}$$

$$\downarrow(1)$$

$$\left[\begin{array}{ccc:c}3&5&6&0\\0&\boxed{2/3}&1&0\\0&1/3&1&1\end{array}\right]m_{32}=1/2$$

$$\downarrow(2)$$

$$\left[\begin{array}{ccc:c}3&5&6&0\\0&2/3&1&0\\0&0&1/2&1\end{array}\right],$$

即

$$x_3=2,\quad x_2=-3,\quad x_1=1.$$

所以,原方程组的解为 $\boldsymbol{x}=(1,-3,2)^{\mathrm{T}}$.

例 5 用 Gauss 列主元消元法解方程组 $\boldsymbol{Ax}=\boldsymbol{b}$,其中

$$\boldsymbol{A}=\begin{bmatrix}3&-2&-2\\1&-2/3&-1/3\\-1&-4&1\end{bmatrix},\quad \boldsymbol{b}=(1,0,-10)^{\mathrm{T}}.$$

解

$$\left[\begin{array}{ccc:c}\boxed{3}&-2&-2&1\\1&-2/3&-1/3&0\\-1&-4&1&-10\end{array}\right]\begin{matrix}\\m_{21}=1/3\\m_{31}=-1/3\end{matrix}$$

$$\downarrow(1)$$

$$\left[\begin{array}{ccc:c}3&-2&-2&1\\0&0&1/3&1/3\\0&\boxed{-14/3}&1/3&29/3\end{array}\right].$$

$$\downarrow r_2\leftrightarrow r_3$$

$$\left[\begin{array}{ccc:c}3&-2&-2&1\\0&-14/3&1/3&-29/3\\0&0&1/3&-1/3\end{array}\right]$$

即

$$x_3=-1,\quad x_2=2,\quad x_1=1.$$

所以,原方程组的解为 $\boldsymbol{x}=(1,2,-1)^{\mathrm{T}}$.

3.3.1.2 Gauss 列主元消元法的计算步骤

分析第 k 次列主元消元，即进行如下初等变换

$$(A^{(k)} | b^{k}) \to (A^{k+1} | b^{(k+1)})$$

$$\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1k}^{(1)} & \cdots & a_{1n}^{(1)} & \vdots & b_1^{(1)} \\ & a_{22}^{(2)} & \cdots & a_{2k}^{(2)} & \cdots & a_{2n}^{(2)} & \vdots & b_2^{(2)} \\ & & \ddots & \vdots & & \vdots & \vdots & \vdots \\ & & & a_{kk}^{(k)} & \cdots & a_{kn}^{(k)} & \vdots & b_k^{(k)} \\ & & & a_{k+1,k}^{(k)} & \cdots & a_{k+1,n}^{(k)} & \vdots & b_{k+1}^{(k)} \\ & & & \vdots & & \vdots & \vdots & \vdots \\ & & & a_{nk}^{(k)} & \cdots & a_{nn}^{(k)} & & b_n^{(k)} \end{bmatrix}$$

$$\downarrow$$

$$\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1k}^{(1)} & a_{1,k+1}^{(1)} & \cdots & a_{1n}^{(1)} & \vdots & b_1^{(1)} \\ & a_{22}^{(2)} & \cdots & a_{2k}^{(2)} & a_{2,k+1}^{(2)} & \cdots & a_{2n}^{(2)} & \vdots & b_2^{(2)} \\ & & \ddots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ & & & a_{kk}^{(k)} & a_{k,k+1}^{(k)} & \cdots & a_{kn}^{(k)} & \vdots & b_k^{(k)} \\ & & & 0 & a_{k+1,k+1}^{(k+1)} & \cdots & a_{k+1,n}^{(k+1)} & \vdots & b_{k+1}^{(k+1)} \\ & & & \vdots & \vdots & & \vdots & & \vdots \\ & & & 0 & a_{n,k+1}^{(k+1)} & \cdots & a_{nn}^{(k+1)} & \vdots & b_n^{(k+1)} \end{bmatrix}$$

上述变换包括选主元和消元两个步骤，因此列主元消元法解线性方程组的计算步骤可归纳如下：

(1)令 $k=1,2,\cdots,n-1$，做下面 1°至 2°步：

1° 选主元步.

$\max\limits_{k\leqslant i\leqslant n} |a_{ik}^{(k)}| = |a_{i_k,k}^{(k)}|$，即主元为第 i_k 行，第 k 列的元素 $a_{i_k,k}$.

若 $i_k=k$，则转消元步 2°；

若 $i_k\neq k$，则先换行，即

$$a_{kj}^{(k)} \leftrightarrow a_{i_k,j}^{(k)}, \quad j=k,\cdots,n,$$
$$b_k^{(k)} \leftrightarrow b_{i_k}^{(k)},$$

再转消元步 2°.

2° 消元步(同 Gauss 顺序消元法).

$$\begin{cases} m_{ik}=a_{ik}^{(k)}/a_{kk}^{(k)}, & i=k+1,k+2,\cdots,n; \\ a_{ij}^{(k+1)}=a_{ij}^{(k)}-m_{ik}a_{kj}^{(k)}, & i,j=k+1,k+2,\cdots,n; \\ b_i^{(k+1)}=b_i^{(k)}-m_{ik}b_k^{(k)}, & i=k+1,k+2,\cdots,n. \end{cases}$$

(2)回代步(同 Gauss 顺序消元法).

$$\begin{cases} x_n = b_n^{(n)}/a_{nn}^{(n)}, \\ x_i = (b_i^{(i)} - \sum\limits_{j=i+1}^{n} a_{ij}^{(i)}x_j)/a_{ii}^{(i)}, \quad i = n-1,n-2,\cdots,1. \end{cases}$$

3.3.1.3 Gauss 列主元消元法算法设计

在 Gauss 消元法及 Gauss 列主元消元法中，若要保留逐次消元过程中的中间数据，则需要很大的存储量，这些中间数据有：

n 个矩阵 $\boldsymbol{A}^{(1)},\boldsymbol{A}^{(2)},\cdots,\boldsymbol{A}^{(n)}$；

n 个右端向量 $\boldsymbol{b}^{(1)},\boldsymbol{b}^{(2)},\cdots,\boldsymbol{b}^{(n)}$；

一个解向量 x.

事实上，没有必要保留中间过程的数据. 在算法设计中，通常采用动态存储的方式，即消元和回代过程的数据存储单元只需要一个矩阵 $\boldsymbol{A}$ 和一个右端向量 $\boldsymbol{b}$ 就够了.

Algo. 3.1(Gauss 列主元消元法)

1° 选主元步，即 $|a_{i_k,k}|=\max\limits_{k\leqslant i\leqslant n}|a_{ik}|$.

2° 若 $a_{i_k,k}=0$, stop(输出失败标志).

3° 若 $i_k=k$，则转 4°；否则执行下列换行计算：

$$\begin{cases}a_{kj}\leftrightarrow a_{i_k,j}, & j=k,k+1,\cdots,n,\\ b_k\leftrightarrow b_{i_k}.\end{cases}$$

4° 消元步，即

$$\begin{cases}a_{ik}\leftarrow m_{ik}=a_{ik}/a_{kk}, & i=k+1,k+2,\cdots,n;\\ a_{ij}\leftarrow a_{ij}-a_{ik}a_{kj}, & i,j=k+1,k+2,\cdots,n;\\ b_i\leftarrow b_i-a_{ik}b_k, & i=k+1,k+2,\cdots,n.\end{cases}$$

5° 回代步，即

$$b_n\leftarrow b_n/a_{nn},$$

$$b_i\leftarrow(b_i-\sum_{j=i+1}^{n}a_{ij}b_j)/a_{ii},\quad i=n-1,n-2,\cdots,1.$$

6° 打印 $b_i(i=1,\cdots,n)$, Stop.

注 对 $k=1,2,\cdots,n-1$，做上面 1°至 4°步.

Algo. 3.2(多功能 Gauss 列主元消元法)

1° det←1，对 $k=1,2,\cdots,n-1$，做至第 7°步.

2° 选主元，即

$$|a_{i_k,k}|=\max_{k\leqslant i\leqslant n}|a_{ik}|.$$

3° 若 $a_{i_k,k}=0$，则 det←0, stop.

4° 若 $i_k=k$，则转 5°；否则执行下列运算后再转 5°. 换行

$$a_{kj}\leftrightarrow a_{i_k,j},\quad j=k,\cdots,n,$$

$$b_k\leftrightarrow b_{i_k},$$

$$\det\leftarrow-\det.$$

5° 计算 m_{ik}，且

$$a_{ik}\leftarrow m_{ik}=a_{ik}/a_{kk},\quad i=k+1,k+2,\cdots,n.$$

6° 消元计算

$$a_{ij}\leftarrow a_{ij}-a_{ik}a_{kj},\qquad i,j=k+1,k+2,\cdots,n;$$

$$b_i \leftarrow b_i - a_{ik}b_k, \qquad i=k+1,k+2,\cdots,n.$$

7° $\det \leftarrow a_{kk}\det$.

8° 回代求解

$$b_n \leftarrow b_n/a_{nn},$$

$$b_i \leftarrow (b_i - \sum_{j=i+1}^{n} a_{ij}b_j)/a_{ii}, \quad i=n-1,n-2,\cdots,1.$$

9° $\det \leftarrow a_{nn}\det$.

定理 3.2(Gauss 列主元消元法的条件) 设 $Ax=b$,令 $A^{(1)}=A, b^{(1)}=b$,则

$$(A^{(1)}|b^{(1)}) \xrightarrow[\text{列主元消元}]{n-1\text{ 次}} (A^{(n)}|b^{(n)})$$

的必要充分条件是

$$a_{i_k,k}^{(k)} \neq 0, \quad k=1,2,\cdots,n-1.$$

3.3.2 高斯－若当(Gauss－Jordan)消元法

3.3.2.1 问题的提出

如前所述,Gauss 消元法是通过消元步骤将原方程组化为三角方程组,再通过回代步骤求三角方程组的解,即原方程组的解.其计算过程归纳如下:

$$(A^{(1)}|b^{(1)}) \xrightarrow[\text{消元}]{n-1\text{ 次}} (A^{(n)}|b^{(n)}),$$

其中

$$A^{(n)} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ & & \ddots & \vdots \\ & & & \ddots \\ & & & a_{nn}^{(n)} \end{bmatrix},$$

$$b^{(n)} = (b_1^{(1)}, b_2^{(2)}, \cdots, b_n^{(n)})^{\mathrm{T}}.$$

$$A^{(n)}x = b^{(n)} \xrightarrow{\text{回代}} \begin{cases} x_n = b_n^{(n)}/a_{nn}^{(n)}, \\ x_i = (b_i - \sum_{j=i+1}^{n} a_{ij}^{(i)}x_j)/a_{ii}^{(i)}, i=n-1,n-2,\cdots,1. \end{cases}$$

而 Gauss－Jordan 消元法则取消了回代步骤,直接由消元步骤得到原方程组的解,即

$$(A^{(1)}|b^{(1)}) \xrightarrow[\text{消元}]{n\text{ 次}} (A^{(n)}|b^{(n)}).$$

其中,$A^{(n)}=I$,此时

$$A^{(n)}x=b^{(n)}$$

的解就是 $x=b^{(n)}$.也就是说,Gauss－Jordan 消元只通过消元步就可完成求解任务,因而取消了回代步.

例 6 用 Gauss－Jordan 消元法解 3.1.1 中例 4 的方程组.

解: $(A,b)=\begin{bmatrix}1&2&3&\vdots&1\\2&4&5&\vdots&0\\\boxed{3}&5&6&\vdots&0\end{bmatrix}\xrightarrow{r_1\leftrightarrow r_3}\begin{bmatrix}3&5&6&\vdots&0\\2&4&5&\vdots&0\\1&2&3&\vdots&1\end{bmatrix}\xrightarrow{(1)}\begin{bmatrix}1&5/3&2&\vdots&0\\0&\boxed{2/3}&1&\vdots&0\\0&1/3&1&\vdots&1\end{bmatrix}$

$$\xrightarrow{(2)}\begin{bmatrix}1&0&-1/2&\vdots&0\\0&1&3/2&\vdots&0\\0&0&\boxed{1/2}&\vdots&1\end{bmatrix}\longrightarrow\begin{bmatrix}1&0&0&\vdots&1\\0&1&0&\vdots&3\\0&0&1&\vdots&2\end{bmatrix}.$$

所以,原方程组的解为 $x=(1,-3,2)^{\mathrm{T}}$,与 Guass 列主元消元法求得的该方程组的解相同.

3.3.2.2 用 Gauss−Jordan 列主元消元法求矩阵的逆阵

分析例 6 中的方程组 $Ax=b$,其中 $b=(1,0,0)^{\mathrm{T}}=e_1$.

用 Gauss−Jordan 列主元消元法求得该方程组的解为

$$x=(1,-3,2)^{\mathrm{T}}.$$

容易验证该解实际上就是矩阵 A 的逆阵 A^{-1} 的第 1 列 A^{-1}_{*1},即

$$x=A^{-1}e_1=[A^{-1}_{*1},A^{-1}_{*2},A^{-1}_{*3}]\begin{bmatrix}1\\0\\0\end{bmatrix}=A^{-1}_{*1}.$$

同理,用 Gauss−Jordan 列主元消元法分别解方程组 $Ax=e_2$ 及 $Ax=e_3$,所得到的解分别为 A^{-1} 的第 2 列及第 3 列,即 A^{-1}_{*2},A^{-1}_{*3}.

注 用 A_{i*} 代表矩阵 A 的第 i 行;用 A_{*j} 代表矩阵 A 的第 j 列. 以下同.

上述求解过程可图示为

$$[A|e_1]\xrightarrow[\text{G-J 消元法}]{}[I_3|A^{-1}_{*1}],$$

$$[A|e_2]\xrightarrow[\text{G-J 消元法}]{}[I_3|A^{-1}_{*2}],$$

$$[A|e_3]\xrightarrow[\text{G-J 消元法}]{}[I_3|A^{-1}_{*3}].$$

综上可得

$$[A|e_1,e_2,e_3]\xrightarrow[\text{G-J 消元法}]{}[I_3|A^{-1}].$$

定理 3.3 设 $A=(a_{ij})_{n\times n}$,$\det(A)\neq0$. 若

$$[A|I_n]\xrightarrow[\text{G-J 消元法}]{}[I_n|T],$$

则

$$T=A^{-1}.$$

例 7 求矩阵 $A=\begin{bmatrix}1&2&3\\2&4&5\\3&5&6\end{bmatrix}$ 的逆矩阵 A^{-1}.

解 用 Gauss−Jordan 列主元消元法求 A 的逆矩阵 A^{-1}:

$$\begin{bmatrix}1&2&3&\vdots&1&0&0\\2&4&5&\vdots&0&1&0\\\boxed{3}&5&6&\vdots&0&0&1\end{bmatrix}\xrightarrow{r_1\leftrightarrow r_3}\begin{bmatrix}3&5&6&\vdots&0&0&1\\2&4&5&\vdots&0&1&0\\1&2&3&\vdots&1&0&0\end{bmatrix}$$

$$\xrightarrow{(1)}\left[\begin{array}{ccc:ccc}1 & 5/3 & 2 & 0 & 0 & 1/3\\0 & \boxed{2/3} & 1 & 0 & 1 & -2/3\\0 & 1/3 & 1 & 1 & 0 & -1/3\end{array}\right]\longrightarrow\left[\begin{array}{ccc:ccc}1 & 0 & -1/2 & 0 & -5/2 & 2\\0 & 1 & 3/2 & 0 & 3/2 & -1\\0 & 0 & \boxed{1/2} & 1 & -1/2 & 0\end{array}\right]$$

$$\longrightarrow\left[\begin{array}{ccc:ccc}1 & 0 & 0 & 1 & -3 & 2\\0 & 1 & 0 & -3 & 3 & -1\\0 & 0 & 1 & 2 & -1 & 0\end{array}\right],$$

所以
$$A^{-1}=\begin{bmatrix}1 & -3 & 2\\-3 & 3 & -1\\2 & -1 & 0\end{bmatrix}.$$

3.4 矩阵的三角分解

通过用矩阵的三角分解来求解线性方程组的方法也是一种直接法. 其基本思想是将系数矩阵分解为两个三角阵,从而将原方程组分解为两个三角方程组,通过求解两个三角方程组获得原方程组的解.

针对不同类型的系数矩阵,我们介绍两种常用的三角分解算法:一种是当系数矩阵 $\boldsymbol{A}$ 为一般稠密矩阵的杜利特尔(Doolittle)分解,由此形成 $\boldsymbol{LU}$ 分解算法;一种是当 $\boldsymbol{A}$ 为三对角矩阵(带型稀疏矩阵)的克劳特(Crout)分解,由此形成追赶算法.

为了推导 $\boldsymbol{LU}$ 分解定理,作为预备知识,下面引入两种初等变换矩阵,即初等排列阵和佛罗伯尼欧斯(Frobenius)矩阵.

3.4.1 初等变换矩阵

3.4.1.1 初等排列阵及其性质

设 $\boldsymbol{I}$ 为 n 阶单位矩阵,$\boldsymbol{I}=[e_1,\cdots,e_i,\cdots,e_j,\cdots,e_n]$,则称 $\boldsymbol{I}_{i,j}=[e_1,\cdots,e_j,\cdots,e_i,\cdots,e_n]$ 为初等排列阵(或置换阵).

初等排列阵 $\boldsymbol{I}_{i,j}$ 有如下性质:

(1) $\boldsymbol{I}_{i,j}=\boldsymbol{I}_{i,j}^{\mathrm{T}}=\boldsymbol{I}_{i,j}^{-1}$(读者自己验证).

(2)设 $\boldsymbol{A}$ 为 n 阶方阵,按列或按行将其表示为

$$\boldsymbol{A}=[\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_n]=\begin{bmatrix}\boldsymbol{a}^{(1)^{\mathrm{T}}}\\\boldsymbol{a}^{(2)^{\mathrm{T}}}\\\vdots\\\boldsymbol{a}^{(n)^{\mathrm{T}}}\end{bmatrix}.$$

其中,$\boldsymbol{a}_i=(a_{1i},a_{2i},\cdots,a_{ni})^{\mathrm{T}}$,$\boldsymbol{a}^{(i)^{\mathrm{T}}}=(a_{i1},a_{i2},\cdots,a_{in})$,则

$$\boldsymbol{A}\boldsymbol{I}_{i,j}=[\boldsymbol{a}_1,\cdots,\underset{(i)}{\boldsymbol{a}_j},\cdots,\underset{(j)}{\boldsymbol{a}_i},\cdots,\boldsymbol{a}_n],$$

$$
I_{i,j}A=\begin{bmatrix} a^{(1)^T} \\ \vdots \\ a^{(j)^T} \\ \vdots \\ a^{(i)^T} \\ \vdots \\ a^{(n)^T} \end{bmatrix}\begin{matrix} \\ \\ (i) \\ \\ (j) \\ \\ \\ \end{matrix}.
$$

即是说，用 $I_{i,j}$ 右乘 A，并将 A 的第 i 列与第 j 列对换；再用 $I_{i,j}$ 左乘 A，再将 A 的第 i 行与第 j 行对换(读者自己验证).

3.4.1.2　Frobenius 矩阵及其性质

设 $A^{(k)}$ 是对 A 进行 $k-1$ 次消元的结果，即

$$
A^{(k)}=\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1k}^{(1)} & \cdots & a_{1n}^{(1)} \\ & a_{22}^{(2)} & \cdots & a_{2k}^{(2)} & \cdots & a_{2n}^{(2)} \\ & & \ddots & \vdots & & \vdots \\ & & & a_{kk}^{(k)} & \cdots & a_{kn}^{(k)} \\ & & & a_{k+1,k}^{(k)} & \cdots & a_{k+1,n}^{(k)} \\ & & & \vdots & & \vdots \\ & & & a_{nk}^{(k)} & \cdots & a_{nn}^{(k)} \end{bmatrix},
$$

而 $m_{ik}=a_{ik}^{(k)}/a_{kk}^{(k)}$，$i=k+1,k+2,\cdots,n$，则称

$$
F_k=\begin{bmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & & \vdots \\ \vdots & & 1 & \vdots & & \vdots \\ \vdots & & -m_{k+1,k} & 1 & & \vdots \\ \vdots & & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & -m_{nk} & 0 & \cdots & 1 \end{bmatrix} \tag{3-6}
$$

为 $A^{(k)}$ 的 n 阶初等下三角阵(简称 F 阵). 例如

$$
F_1=\begin{bmatrix} 1 & & & \\ -m_{21} & 1 & & \\ \vdots & \vdots & \ddots & \\ -m_{n1} & 0 & \cdots & 1 \end{bmatrix}
$$

是 $A^{(1)}=A$ 的 n 阶 F 阵，而

$$
F_2=\begin{bmatrix} 1 & & & & & \\ 0 & 1 & & & & \\ 0 & -m_{32} & 1 & & & \\ \vdots & \vdots & 0 & \ddots & & \\ \vdots & \vdots & \vdots & & \ddots & \\ 0 & -m_{n2} & 0 & \cdots & \cdots & 1 \end{bmatrix}
$$

是 $A^{(2)}=A$ 的 n 阶 F 阵.

F 阵具有如下性质：

(1) F_k 阵的逆阵 F_k^{-1} 为

$$F_k^{-1}=\begin{bmatrix}1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & m_{k+1,k} & \ddots & & \\ & & \vdots & & \ddots & \\ & & m_{nk} & & & 1\end{bmatrix}; \tag{3-7}$$

(2) $F_kA^{(k)}=A^{(k+1)}$，例如 $F_1A^{(1)}=A^{(2)}$. 事实上

$$F_1A^{(1)}=\begin{bmatrix}1 & & & \\ -m_{21} & 1 & & \\ \vdots & \vdots & \ddots & \\ -m_{n1} & 0 & \cdots & 1\end{bmatrix}\begin{bmatrix}a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} \\ \vdots & \vdots & & \vdots \\ a_{n1}^{(1)} & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)}\end{bmatrix}$$

$$=\begin{bmatrix}a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)}\end{bmatrix}=A^{(2)}.$$

同理

$$F_2A^{(2)}=A^{(3)},\cdots,F_{n-1}A^{(n-1)}=A^{(n)}. \tag{3-8}$$

3.4.2 矩阵的 LU 分解定理

3.4.2.1 矩阵的三角化

引理 3.2 设 $A=(a_{ij})_{n\times n}$，且 $\det(A_i)\neq 0, i=1,2,\cdots,n-1$，则惟一存在 $n-1$ 个矩阵 $F_k, k=1,2,\cdots,n-1$，使

$$F_{n-1}F_{n-2}\cdots F_1A=A^{(n)}.$$

证 由引理 3.1 及定理 3.1 知，当 $\det(A_i)\neq 0, i=1,2,\cdots,n-1$ 时，有 $a_{ii}^{(i)}\neq 0, i=1,2,\cdots,n-1$，故 $n-1$ 次消元所作的辅助乘数

$$m_{ik}=a_{ik}^{(k)}/a_{kk}^{(k)},\quad k=1,2,\cdots,n-1,\quad i=k+1,\cdots,n$$

存在，由它们所构造的 $F_1,F_2,\cdots,F_{n-1}$ 也存在，即有

$$F_1=\begin{bmatrix}1 & & & \\ m_{21} & 1 & & \\ \vdots & \vdots & \ddots & \\ -m_{n1} & 0 & \cdots & 1\end{bmatrix},\quad F_2=\begin{bmatrix}1 & & & & \\ 0 & 1 & & & \\ 0 & -m_{32} & 1 & & \\ \vdots & \vdots & \vdots & \ddots & \\ 0 & -m_{n2} & 0 & \cdots & 1\end{bmatrix},\cdots,$$

$$
F_{n-1}=\begin{bmatrix} 1 & & & & \\ 0 & 1 & & & \\ \vdots & \vdots & \ddots & & \\ \vdots & \vdots & & 1 & \\ 0 & 0 & \cdots & -m_{n,n-1} & 1 \end{bmatrix}.
$$

又由 F 阵的性质 2，即式(3－8)，有

$$
F_1A^{(1)}=A^{(2)}, F_2A^{(2)}=A^{(3)}, \cdots, F_{n-1}A^{(n-1)}=A^{(n)},
$$

因而有

$$
F_{n-1}F_{n-2}\cdots F_1A=A^{(n)}.
$$

即是说，对 A 连续左乘 $n-1$ 个 F 阵便可得到一个上三角阵 $A^{(n)}$，且

$$
A^{(n)}=\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ & & \ddots & \vdots \\ & & & \ddots \\ & & & a_{nn}^{(n)} \end{bmatrix}.
$$

3.4.2.2 *LU* 分解定理

定理 3.4 设 $A=(a_{ij})_{n\times n}$，惟一存在矩阵 L,U，使得 $A=LU$ 的充要条件是

$$
\det(A_i)\neq 0,\quad i=1,2,\cdots,n-1.
$$

其中

$$
L=\begin{bmatrix} 1 & & & \\ l_{21} & 1 & & \\ l_{31} & l_{32} & \ddots & \\ \vdots & \vdots & & \ddots \\ l_{n1} & l_{n2} & \cdots & 1 \end{bmatrix}=\begin{bmatrix} 1 & & & \\ m_{21} & 1 & & \\ m_{31} & m_{32} & \ddots & \\ \vdots & \vdots & & \ddots \\ m_{n1} & m_{n2} & \cdots & 1 \end{bmatrix},
$$

$$
U=\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ & u_{22} & \cdots & u_{2n} \\ & & \ddots & \vdots \\ & & & u_{nn} \end{bmatrix}=\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ & & \ddots & \vdots \\ & & & a_{nn}^{(n)} \end{bmatrix}=A^{(n)}.
$$

证 首先证存在性：

由题设 $\det(A_i)\neq 0, i=1,2,\cdots,n-1$ 及引理 3.2，则存在 $F_1,F_2,\cdots,F_{n-1}$ 使

$$
F_{n-1}F_{n-2}\cdots F_1A=A^{(n)},
$$

即

$$
A=F_1^{-1}F_2^{-1}\cdots F_{n-1}^{-1}A^{(n)}.
$$

令

$$
L=F_1^{-1}F_2^{-1}\cdots F_{n-1}^{-1},
$$

则

$$
A=L\cdot A^{(n)}=LU.
$$

容易验证

$$
L=F_1^{-1}F_2^{-1}\cdots F_{n-1}^{-1}=(F_{n-1}F_{n-2}\cdots F_2F_1)^{-1}
$$

$$
=\begin{bmatrix} 1 & & & & \\ m_{21} & 1 & & & \\ m_{31} & m_{32} & \ddots & & \\ \vdots & \vdots & & \ddots & \\ m_{n1} & m_{n2} & \cdots & \cdots & 1 \end{bmatrix}.
$$

其次证惟一性：

相反设存在 L_1, U_1 及 L_2, U_2 使得

$$A=L_1U_1=L_2U_2,$$

则有
$$L_2^{-1}(L_1U_1)U_1^{-1}=L_2^{-1}(L_2U_2)U_1^{-1},$$

即
$$L_2^{-1}L_1=U_2U_1^{-1}.$$

式中，左边 $L_2^{-1}L_1$ 为单位下三角阵，右边 $U_2U_1^{-1}$ 为上三角阵. 由此

$$L_2^{-1}L_1=U_2U_1^{-1}=I_n,$$

即
$$L_2^{-1}=L_1^{-1}\longrightarrow L_2=L_1,$$

$$U_1^{-1}=U_2^{-1}\longrightarrow U_1=U_2.$$

故惟一性得证.

下面举例说明矩阵进行 LU 分解的条件.

例 8 设 $A=\begin{bmatrix}1 & 2 & 6\\ 2 & 5 & 15\\ 6 & 15 & 46\end{bmatrix}$，因为

$$\det(A_1)=1,\quad \det(A_2)=1,\quad \det(A_3)=1,$$

据定理 3.4，存在惟一的 L, U 阵，使得 $A=LU$，其中

$$L=\begin{bmatrix}1 & 0 & 0\\ 2 & 1 & 0\\ 6 & 3 & 1\end{bmatrix},\quad U=\begin{bmatrix}1 & 2 & 6\\ 0 & 1 & 3\\ 0 & 0 & 1\end{bmatrix}.$$

例 9 设 $A=\begin{bmatrix}1 & -1/2 & -1/2\\ -1/2 & 1 & -1/2\\ -1/2 & -1/2 & 1\end{bmatrix}$，因为

$$\det(A_1)\neq 0,\quad \det(A_2)\neq 0,\quad \det(A_3)=0,$$

据定理 3.4 仍存在惟一的 L, U 矩阵，使得 $A=LU$，其中

$$L=\begin{bmatrix}1 & 0 & 0\\ -1/2 & 1 & 0\\ -1/2 & -1 & 1\end{bmatrix},\quad U=\begin{bmatrix}1 & -1/2 & -1/2\\ 0 & 3/4 & 3/4\\ 0 & 0 & 0\end{bmatrix}.$$

例 10 设 $A=\begin{bmatrix}1 & 2 & 3\\ 2 & 4 & 1\\ 4 & 6 & 7\end{bmatrix}$，因为

$$\det(A_1)\neq 0,\quad \det(A_2)=0,$$

所以，不满足定理 3.4 的条件，即不存在 L, U 矩阵使得 $A=LU$.

例 11 设 $A=\begin{bmatrix}1&1&1\\2&2&1\\3&3&1\end{bmatrix}$,因为

$$\det(A_1)\neq 0,\quad \det(A_2)=0,$$

所以,不满足定理 3.4 的条件,但 A 仍可分解,只是分解不是惟一的.例如存在

$$L_1=\begin{bmatrix}1&0&0\\2&1&0\\3&1&1\end{bmatrix},\quad U_1=\begin{bmatrix}1&1&1\\0&0&-1\\0&0&-1\end{bmatrix};$$

$$L_2=\begin{bmatrix}1&0&0\\2&1&0\\3&2&1\end{bmatrix},\quad U_2=\begin{bmatrix}1&1&1\\0&0&-1\\0&0&0\end{bmatrix}.$$

对于本例中的矩阵 A,利用后面 4.3 中的 LU 分解算法,我们可以列出无穷多对 L,U 矩阵,其乘积为 A.

例 10 及例 11 中的矩阵均不满足 LU 分解条件,故均不能惟一地分解.其中例 10 中的矩阵 A 不能分解,而例 11 中的矩阵 A 能分解,但不是惟一的.

3.4.2.3 Crout 分解

将 LU 分解换一个提法:要求 L 为下三角形矩阵,U 为单位上三角形矩阵,即是 Crout 分解.

推论 3.1 设 $A=(a_{ij})_{n\times n}$满足 LU 分解的条件:$\det(A_i)\neq 0, i=1,2,\cdots,n-1$,则惟一存在 $\widetilde{L}$ 与 $\widetilde{U}$,使得

$$A=\widetilde{L}\widetilde{U}.$$

其中 $\widetilde{L}$ 是一个下三角形矩阵,$\widetilde{U}$是一个单位上三角形矩阵.

证 据定理 3.4,$A=LU$,其中

$$L=\begin{bmatrix}1&&&\\l_{21}&1&&\\\vdots&\vdots&\ddots&\\l_{n1}&l_{n2}&\cdots&1\end{bmatrix},$$

$$U=\begin{bmatrix}u_{11}&u_{12}&\cdots&u_{1n}\\&u_{22}&\cdots&u_{2n}\\&&\ddots&\vdots\\&&&u_{nn}\end{bmatrix}=\begin{bmatrix}u_{11}&&&\\&u_{22}&&\\&&\ddots&\\&&&u_{nn}\end{bmatrix}\begin{bmatrix}1&u_{12}/u_{11}&\cdots&u_{1n}/u_{11}\\&1&\cdots&u_{2n}/u_{22}\\&&\ddots&\vdots\\&&&1\end{bmatrix}=D\cdot\widetilde{U}.$$

令

$$\widetilde{L}=LD=\begin{bmatrix}u_{11}&&&\\u_{11}l_{21}&u_{22}&&\\\vdots&\vdots&\ddots&\\u_{11}l_{n1}&u_{22}l_{n2}&\cdots&u_{nn}\end{bmatrix},$$

所以

$$A=\widetilde{L}\widetilde{U}.$$

其中 $\widetilde{L}$ 为下三角形矩阵,$\widetilde{U}$ 为单位上三角形矩阵.

又因为 L,U 是惟一存在的，因而 $\widetilde{L},\widetilde{U}$ 也是惟一存在的.

与 Crout 分解对应的算法称为 Crout 算法. 它用于解线性方程组时，其特点是回代时不做除法. 具体应用见 3.5 节的追赶法.

3.4.3 LU 分解算法

3.4.3.1 LU 分解计算公式

设
$$A:\det(A_i)\neq 0,\quad i=1,2,\cdots,n-1.$$
$$A=LU,$$

其中
$$L=\begin{bmatrix}1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ \vdots & \vdots & \ddots & 1 & \\ l_{n1} & l_{n2} & \cdots & l_{n,n-1} & 1\end{bmatrix},\quad U=\begin{bmatrix}u_{11} & u_{12} & \cdots & u_{1n}\\ & u_{22} & \cdots & u_{2n}\\ & & \ddots & \vdots\\ & & & u_{nn}\end{bmatrix}.$$

下面采用并行方案计算 L,U 阵的元素.

令 $r=1,2,\cdots,n$.

(1)计算 U 阵的第 r 行，即 $U_{r\prec}=(0,\cdots,0,u_{rr},\cdots,u_{rn})$，该行元素记为 $u_{rj}(j=r,r+1,\cdots,n)$.

(2)计算 L 阵的第 r 列，即 $L_{\prec r}=(0,\cdots,0,1,l_{r+1,r},\cdots,l_{nr})^{\mathrm{T}}$，该列元素记为 $l_{ir}(i=r+1,r+2,\cdots,n)$.

(3)计算公式为：

当 $r=1$ 时，
$$\begin{cases}u_{1j}=a_{1j}, & j=\overline{1,n},\\ l_{i1}=a_{i1}/u_{11}, & i=\overline{2,n};\end{cases}\tag{3-9}$$

当 $r=2,3,\cdots,n$ 时，
$$\begin{cases}u_{rj}=a_{rj}-\sum\limits_{k=1}^{r-1}l_{rk}u_{kj}, & j=r,r+1,\cdots,n,\\ l_{ir}=(a_{ir}-\sum\limits_{k=1}^{r-1}l_{ik}u_{kr})/u_{rr}, & i=r+1,r+2,\cdots,n.\end{cases}\tag{3-10}$$

上述计算公式可以列在一张表格上进行，如设
$$A=\begin{bmatrix}a_{11} & a_{12} & a_{13} & a_{14}\\ a_{21} & a_{22} & a_{23} & a_{24}\\ a_{31} & a_{32} & a_{33} & a_{34}\\ a_{41} & a_{42} & a_{43} & a_{44}\end{bmatrix}.$$

若将 A 作 L,U 分解，则 L,U 阵中各元素的计算可列表进行(见表 3－1).

表 3-1

$u_{11}=a_{11}$	$u_{12}=a_{12}$	$u_{13}=a_{13}$	$u_{14}=a_{14}$
$l_{21}=a_{21}/u_{11}$	$u_{22}=a_{22}-l_{21}u_{12}$	$u_{23}=a_{23}-l_{21}u_{13}$	$u_{24}=a_{24}-l_{21}u_{14}$
$l_{31}=a_{31}/u_{11}$	$l_{32}=(a_{32}-l_{31}u_{12})/u_{22}$	$u_{33}=a_{33}-l_{31}u_{13}-l_{32}u_{23}$	$u_{34}=a_{34}-l_{31}u_{14}-l_{32}u_{24}$
$l_{41}=a_{41}/u_{11}$	$l_{42}=(a_{42}-l_{41}u_{12})/u_{22}$	$l_{43}=(a_{43}-l_{41}u_{13}-l_{42}u_{23})/u_{33}$	$u_{44}=a_{44}-l_{41}u_{14}-l_{42}u_{24}-l_{43}u_{34}$

3.4.3.2 LU 分解计算公式的推导

设

$$A=LU,$$

其中

$$L=\begin{bmatrix}1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ l_{n1} & l_{n2} & l_{n3}\cdots & l_{n,n-1} & 1\end{bmatrix},\quad U=\begin{bmatrix}u_{11} & u_{12} & \cdots & u_{1n}\\ & u_{22} & \cdots & u_{2n}\\ & & \ddots & \vdots\\ & & & u_{nn}\end{bmatrix}.$$

(1)U_{1*} 及 L_{*1} 的元素计算公式推导.

因为

$$a_{1j}=L_{1*}U_{*j}=(1,0,\cdots,0)\begin{bmatrix}u_{1j}\\ \vdots\\ u_{jj}\\ 0\\ \vdots\\ 0\end{bmatrix}=u_{1j},$$

所以

$$u_{1j}=a_{1j},\quad j=1,2,\cdots,n.$$

此即计算 U_{1*} 的元素的公式.

又因为

$$a_{i1}=L_{i*}U_{*1}=(l_{i1},\cdots,l_{i,i-1},1,0,\cdots,0)\begin{bmatrix}u_{11}\\ 0\\ \vdots\\ 0\end{bmatrix}=l_{i1}u_{11},$$

所以

$$l_{i1}=a_{i1}/u_{11},\quad i=2,3,\cdots,n.$$

此即计算 L_{*1} 的元素的公式.

综上得到

$$\begin{cases}u_{1j}=a_{1j}, & j=1,2,\cdots,n,\\ l_{i1}=a_{i1}/u_{11}, & i=2,3,\cdots,n.\end{cases}$$

(2)U_{r*} 及 $L_{*r}(r=2,3,\cdots,n)$ 的元素计算公式的推导.

由 $a_{rj}=L_{r*}U_{*j}$ 可导出计算 U_{r*} 的元素的公式

$$u_{rj}=a_{rj}-\sum_{k=1}^{r-1}l_{rk}u_{kj},\quad j=r,r+1,\cdots,n,$$

由 $a_{ir}=L_{i*}U_{*r}$ 可导出计算 L_{*r} 的元素的计算公式

$$l_{ir}=(a_{ir}-\sum_{k=1}^{r-1}l_{ik}u_{kr})/u_{rr},\quad i=r+1,r+2,\cdots,n.$$

3.4.3.3 用 *LU* 分解方法解线性方程组

设 $Ax=b, A=LU$. 将 $A=LU$ 代人 $Ax=b$ 中有

$$LUx=b.$$

令 $y=Ux$，则有

$$\begin{cases} Ly=b(\text{下三角方程组}), \\ Ux=y(\text{上三角方程组}). \end{cases}$$

(1)解 $Ly=b$，即

$$\begin{bmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ \vdots & \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & l_{n3} & \cdots & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}.$$

由 $L_{1\text{行}}y=b_1$，即

$$(1,0,\cdots,0)\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}=b_1,$$

得
$$y_1=b_1;$$

由 $L_{i\text{行}}y=b_i$，即

$$(l_{i1},\cdots l_{i,i-1},1,0,\cdots,0)\begin{bmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{bmatrix}=b_i,$$

得
$$\sum_{k=1}^{i-1} l_{ik}\, y_k + y_i = b_i,$$

即
$$y_i = b_i - \sum_{k=1}^{i-1} l_{ik}\, y_k, \quad i=2,3,\cdots,n.$$

综上得
$$\begin{cases} y_1=b_1, \\ y_i = b_i - \sum\limits_{k=1}^{i-1} l_{ik}\, y_k, \quad i=2,3,\cdots,n. \end{cases} \tag{3-11}$$

(2)解 $Ux=y$，即

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ & u_{22} & \cdots & u_{2n} \\ & & \ddots & \vdots \\ & & & u_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}.$$

由 $U_{n\text{行}}x=y_n$，即

$$(0,\cdots,0,u_{nn})\begin{bmatrix}x_1\\x_2\\\vdots\\x_n\end{bmatrix}=y_n,$$

得 $$u_{nn}x_n=y_n,\quad 即\ x_n=y_n/u_{nn};$$

由 $U_{i*}x=y_i$，即

$$(0,\cdots,0,u_{ii},u_{i,i+1},\cdots,u_{in})\begin{bmatrix}x_1\\\vdots\\x_i\\\vdots\\x_n\end{bmatrix}=y_i,$$

得 $$u_{ii}x_i+\sum_{k=i+1}^{n}u_{ik}x_k=y_i,$$

即 $$x_i=(y_i-\sum_{k=i+1}^{n}u_{ik}x_k)/u_{ii},\quad i=n-1,n-2,\cdots,1.$$

综上得
$$\begin{cases}x_n=y_n/u_{nn},\\ x_i=(y_i-\sum\limits_{k=i+1}^{n}u_{ik}x_k)/u_{ii},\quad i=n-1,n-2,\cdots,1.\end{cases}\tag{3-12}$$

例 12 求方程组 $Ax=b$ 的解，其中

$$A=\begin{bmatrix}1&2&3\\2&5&2\\3&1&5\end{bmatrix},\quad b=(14,18,20)^T.$$

解 对 A 进行 LU 分解，即

$$\begin{aligned}&u_{11}=1, && u_{12}=2, && u_{13}=3,\\ &l_{21}=2, && u_{22}=5-4=1, && u_{23}=2-6=-4,\\ &l_{31}=3, && l_{32}=1-6=-5, && u_{33}=5-9-20=-24.\end{aligned}$$

由此可得

$$L=\begin{bmatrix}1&0&0\\2&1&0\\3&-5&1\end{bmatrix},\quad U=\begin{bmatrix}1&2&3\\0&1&-4\\0&0&-24\end{bmatrix}.$$

解下三角方程组 $Ly=b$. 据式(3－11)得

$$y=(14,-10,-72)^T.$$

解上三角方程组 $Ux=y$. 据式(3－12)得

$$x=(1,2,3)^T.$$

例 13 求解方程组 $Ax=b$，其中

$$A=\begin{bmatrix}8.1&2.3&-1.5\\0.5&-6.23&0.87\\2.5&1.5&10.2\end{bmatrix},\quad b=(6.1,2.3,1.8)^T.$$

计算过程中保留4位有效数字.

解 对 A 进行 LU 分解,即

$$u_{11}=8.100,\quad u_{12}=2.300,\quad u_{13}=-1.500,$$
$$l_{21}=0.06173,\quad u_{22}=-6.372,\quad u_{23}=0.9626,$$
$$l_{31}=0.3086,\quad l_{32}=-0.1240,\quad u_{33}=10.78.$$

由此可得

$$L=\begin{bmatrix}1.000 & 0 & 0\\ 0.06173 & 1.000 & 0\\ 0.3086 & -0.1240 & 1.000\end{bmatrix},\quad U=\begin{bmatrix}8.100 & 2.300 & -1.500\\ 0 & -6.372 & 0.9626\\ 0 & 0 & 10.78\end{bmatrix}.$$

解下三角方程组 $Ly=b$. 据式(3-11)得

$$y=(6.100,1.9230,0.1560)^{\mathrm T}.$$

解上三角方程组 $Ux=y$. 据式(3-12)得

$$x=(0.8408,-0.2996,0.01447)^{\mathrm T}.$$

3.4.3.4 *选主元的 LU 分解(按列选主元)*

例 14 求解方程组 $Ax=b$,其中

对 A 进行 LU 分解:

$$A=\begin{bmatrix}0 & 3 & 4\\ 1 & -1 & 1\\ 2 & 1 & 2\end{bmatrix},\quad b=(1,0,0)^{\mathrm T}.$$

解 矩阵 A 不满足 LU 分解的条件,但我们可以通过换行后再进行分解,即

$$A=\begin{bmatrix}0 & 3 & 4\\ 1 & -1 & 1\\ 2 & 1 & 2\end{bmatrix}\xrightarrow{r_1\leftrightarrow r_3}\begin{bmatrix}2 & 1 & 2\\ 1 & -1 & 1\\ 0 & 3 & 4\end{bmatrix}.$$

对 A 进行 LU 分解:

$$u_{11}=2,\quad u_{12}=1,\quad u_{13}=2,$$
$$l_{21}=1/2,\quad a_{22}=-1,\quad a_{23}=1,$$
$$l_{31}=0,\quad a_{32}=3,\quad a_{33}=4.$$

$$\downarrow r_2\leftrightarrow r_3$$

$$u_{11}=2,\quad u_{12}=1,\quad u_{13}=2,$$
$$l_{21}=0,\quad a_{22}=3,\quad a_{23}=4,$$
$$l_{31}=1/2,\quad a_{32}=-1,\quad a_{33}=1.$$

$$\downarrow$$

$$u_{11}=2,\quad u_{12}=1,\quad u_{13}=2,$$
$$l_{21}=0,\quad u_{22}=3,\quad u_{23}=4,$$
$$l_{31}=1/2,\quad l_{32}=-1/2,\quad u_{33}=2.$$

故

$$L=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 1/2 & 1/2 & 1\end{bmatrix},\quad U=\begin{bmatrix}2 & 1 & 2\\ 0 & 3 & 4\\ 0 & 0 & 2\end{bmatrix}.$$

注意，上述分解并非 $A=LU$，而是

$$A \xrightarrow{r_1 \leftrightarrow r_3} B \xrightarrow{r_2 \leftrightarrow r_3} C = \begin{bmatrix} 2 & 1 & 2 \\ 0 & 3 & 4 \\ 1 & -1 & 1 \end{bmatrix} = LU,$$

即

$$I_{2,3} I_{1,3} A = C = LU.$$

若欲解方程组 $Ax=b$，因 $b=(1,0,0)^T$，则

$$Ax=b \Longleftrightarrow I_{2,3} I_{1,3} Ax = I_{2,3} I_{1,3} b,$$

即 $Ax=b$ 与 $Cx=b'=(0,1,0)^T$ 是等价的. 以下求解与例 13 相同.

第一步，解下三角形方程组 $Ly=b'$：

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1/2 & -1/2 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \to y = \begin{bmatrix} 0 \\ 1 \\ 1/2 \end{bmatrix};$$

第二步，解上三角形方程组 $Ux=y$：

$$\begin{bmatrix} 2 & 1 & 2 \\ 0 & 3 & 4 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1/2 \end{bmatrix} \to x = \begin{bmatrix} -1/4 \\ 0 \\ 1/4 \end{bmatrix};$$

所以原方程组 $Ax=b$ 的解为 $x=(-\frac{1}{4},0,\frac{1}{4})^T$.

3.5 追赶法

3.5.1 三对角阵的克劳特(Crout)分解

3.5.1.1 三对角阵

设

$$A = \begin{bmatrix} b_1 & c_1 & & & & & \\ a_2 & b_2 & c_2 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & a_i & b_i & c_i & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & & & a_n & b_n \end{bmatrix},$$

其中，若 $b_i \neq 0, i=1,2,\cdots,n; a_i \neq 0, i=2,3,\cdots,n; c_i \neq 0, i=1,2,\cdots,n-1$，则称矩阵 A 为不可约的 n 阶三对角阵.

3.5.1.2 Crout 分解定理

定理 3.5 设 A 为 n 阶三对角阵，且

(1) A 为不可约，即

$$b_i \neq 0, i=1,2,\cdots,n; \quad a_i \neq 0, i=2,3,\cdots,n; \quad c_i \neq 0, i=1,2,\cdots,n-1.$$

(2) A 的对角元素弱超(弱占优)，即

$$|b_1|>|c_1|>0;$$
$$|b_i|\geqslant|a_i|+|c_i|,\quad i=2,3,\cdots,n-1;$$
$$|b_n|>|a_n|.$$

则惟一存在两个三角阵 $\boldsymbol{L},\boldsymbol{U}$,使 $\boldsymbol{A}=\boldsymbol{L}\boldsymbol{U}$,其中

$$\boldsymbol{L}=\begin{bmatrix}\alpha_1 & & & & & \\ r_2 & \alpha_2 & & & & \\ & \ddots & \ddots & & & \\ & & r_i & \alpha_i & & \\ & & & \ddots & \ddots & \\ & & & & r_n & \alpha_n\end{bmatrix},\quad \boldsymbol{U}=\begin{bmatrix}1 & \beta_1 & & & & \\ & 1 & \beta_2 & & & \\ & & \ddots & \ddots & & \\ & & & 1 & \beta_i & \\ & & & & \ddots & \ddots \\ & & & & 1 & \beta_{n-1} \\ & & & & & 1\end{bmatrix},$$

且 $\boldsymbol{L}$ 为下三角阵,其元素为 $\alpha_i(i=1,2,\cdots,n),r_i(i=2,3,\cdots,n)$;$\boldsymbol{U}$ 为单位上三角阵,其元素为 $\beta_i(i=1,2,\cdots,n-1)$.

3.5.1.3 $\boldsymbol{L},\boldsymbol{U}$ 阵元素的计算公式

$\boldsymbol{L},\boldsymbol{U}$ 阵各元素的计算公式为

$$r_i=a_i,\quad i=2,3,\cdots,n.$$

而 α_i、β_i 的计算按以下并行方案进行(推导见 3.5.3).

$$\xrightarrow{\alpha_1=b_1,\beta_1=c_1/\alpha_1}$$
$$\xrightarrow{\alpha_2=b_2-a_2\beta_1,\quad \beta_2=c_2/\alpha_2}$$
$$\vdots$$
$$\xrightarrow{\alpha_i=b_i-a_i\beta_{i-1},\quad \beta_i=c_i/\alpha_i}$$
$$\vdots$$
$$\xrightarrow{\alpha_{n-1}=b_{n-1}-a_{n-1}\beta_{n-2},\quad \beta_{n-1}=c_{n-1}/\alpha_{n-1}}$$
$$\xrightarrow{\alpha_n=b_n-a_n\beta_n}$$

综上归纳为

$$\begin{aligned}&r_i=a_i, && i=2,3,\cdots,n;\\ &\begin{cases}\alpha_1=b_1,\\ \alpha_i=b_i-a_i\beta_{i-1}, & i=2,3,\cdots,n;\end{cases}\\ &\beta_i=c_i/\alpha_i, && i=1,2,\cdots,n-1.\end{aligned}\tag{3-13}$$

3.5.2 追赶法(利用 Crout 分解解线性方程组)

设三对角方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{f}$,利用追赶法求解线性方程组的步骤如下:

(1)分解

$$\boldsymbol{A}=\boldsymbol{L}\boldsymbol{U},\quad \boldsymbol{A}\boldsymbol{x}=\boldsymbol{f}\begin{cases}\boldsymbol{L}\boldsymbol{y}=\boldsymbol{f},\\ \boldsymbol{U}\boldsymbol{x}=\boldsymbol{y};\end{cases}$$

(2)解下三角方程组 $Ly=f$，计算公式(推导见 3.5.3)为

$$\begin{cases} y_1=f_1/b_1, \\ y_i=(f_i-a_iy_{i-1})/\alpha_i, \quad i=2,3,\cdots,n; \end{cases} \tag{3-14}$$

(3)解上三角方程组 $Ux=y$，计算公式(推导见 3.5.3)为

$$\begin{cases} x_n=y_n, \\ x_i=y_i-\beta_i x_{i+1}, \quad i=n-1,n-2,\cdots,1. \end{cases} \tag{3-15}$$

上述步骤中，我们将计算辅助向量 y 的各分量 $y_1\to y_2\to\cdots\to y_n$ 的过程称为"追"的过程，将计算方程组的解 $x_n\to x_{n-1}\to\cdots\to x_1$ 的过程称为"赶"的过程。由此，综合式(3－13)，(3－14)，(3－15)形成追赶法计算表格：

$\alpha_1=b_1 \to \alpha_2=b_2-a_2\beta_1 \to\cdots\to \alpha_{n-1}=b_{n-1}-a_{n-1}\beta_{n-2} \to \alpha_n=b_n-a_n\beta_{n-1}$

$\beta_1=c_1/\alpha_1 \to \beta_2=c_2/\alpha_2 \to\cdots\to \beta_{n-1}=c_{n-1}/\alpha_{n-1}$

$y_1=\dfrac{f_1}{\alpha_1} \to y_2=\dfrac{f_2-a_2y_1}{\alpha_2} \to\cdots\to y_{n-1}=\dfrac{f_{n-1}-a_{n-1}y_{n-2}}{\alpha_{n-1}} \to y_n=\dfrac{f_n-a_ny_{n-1}}{\alpha_n}$

——————————————追——————————————→

$x_1=y_1-\beta_1x_2 \leftarrow x_2=y_2-\beta_2x_3 \leftarrow\cdots\leftarrow x_{n-1}=y_{n-1}-\beta_{n-1}x_n \leftarrow x_n=y_n$

←——————————————赶——————————————

若再将上述表格加上方程组 $Ax=f$ 的已知数据，即形成完整的追赶法计算表格(见表 3－2)。

表 3－2

i	1	2	…	$n-1$	n
c_i	c_1	c_2	…	c_{n-1}	
b_i	b_1	b_2	…	b_{n-1}	b_n
a_i		a_2	…	a_{n-1}	a_n
f_i	f_1	f_2	…	f_{n-1}	f_n
α_i	$\alpha_1=b_1$	$\alpha_2=b_2-a_2\beta_1$	…	$\alpha_{n-1}=b_{n-1}-a_{n-1}\beta_{n-2}$	$\alpha_n=b_n-a_n\beta_{n-1}$
β_i	$\beta_1=c_1/\alpha_1$	$\beta_2=c_2/\alpha_2$	…	$\beta_{n-1}=c_{n-1}/\alpha_{n-1}$	
y_i	$y_1=f_1/\alpha_1$	$y_2=(f_2-a_2y_1)/\alpha_2$	…	$y_{n-1}=(f_{n-1}-a_{n-1}y_{n-2})/\alpha_{n-1}$	$y_n=(f_n-a_ny_{n-1})/\alpha_n$
x_i	$x_1=y_1-\beta_1x_2$	$x_2=y_2-\beta_2x_3$	…	$x_{n-1}=y_{n-1}-\beta_{n-1}x_n$	$x_n=y_n$

表 3－2 中第二、三、四、五栏是已知数据(系数矩阵及右端项)，第六、七、八栏是 Crout 分解结果 α_i、β_i 及辅助向量 y，第九栏是三对角方程组的解向量 x.

例 15 解线性方程组 $Ax=b$，其中

$$A=\begin{bmatrix} 4 & 1 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 1 & 4 \end{bmatrix}, \quad b=(1,0.5,-1,3,2)^{\mathrm{T}}.$$

要求在计算中保留4位有效数字.

解 据追赶法计算表3－2列出如下计算结果,见表3－3:

表3－3

i	1	2	3	4	5
c_i	1	1	1	1	
b_i	4	4	4	4	4
a_i		1	1	1	1
f_i	1	0.5	－1	3	2
α_i	$\alpha_1=4.000$	$\alpha_2=3.750$	$\alpha_3=3.733$	$\alpha_4=3.732$	$\alpha_5=3.732$
β_i	$\beta_1=0.2500$	$\beta_2=0.2667$	$\beta_3=0.2679$	$\beta_4=0.2680$	
y_i	$y_1=0.2500$	$y_2=0.6667$	$y_3=-0.2858$	$y_4=0.8805$	$y_5=-0.7720$
x_i	$x_1=0.1949$	$x_2=0.2206$	$x_3=-0.5770$	$x_4=1.087$	$x_5=-0.7720$

所以方程组的解为

$$x=(0.1949,0.2206,-0.5770,1.087,-0.7720)^{\mathrm{T}}.$$

3.5.3 追赶法求解公式的推导

设 $A=LU$,即

$$\begin{bmatrix} b_1 & c_1 & & & & & \\ a_2 & b_2 & c_2 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & a_i & b_i & c_i & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & & & a_n & b_n \end{bmatrix}$$

$$=\begin{bmatrix} \alpha_1 & & & & & \\ r_2 & \alpha_2 & & & & \\ & \ddots & \ddots & & & \\ & & r_i & \alpha_i & & \\ & & & \ddots & \ddots & \\ & & & & r_n & \alpha_n \end{bmatrix}\begin{bmatrix} 1 & \beta_1 & & & & & \\ & 1 & \beta_2 & & & & \\ & & \ddots & \ddots & & & \\ & & & 1 & \beta_{i-1} & & \\ & & & & \ddots & \ddots & \\ & & & & & 1 & \beta_{n-1} \\ & & & & & & 1 \end{bmatrix}.$$

(1)公式(3－13)的推导.

因为

$$\underset{(i,i-1)}{\underset{\downarrow}{a_i}} = L_{i*}U_{*i-1} = (0,\cdots,0,\ \underset{(i-1)}{\underset{\downarrow}{r_i}},\ \underset{(i)}{\underset{\downarrow}{\alpha_i}},0,\cdots,0)\begin{bmatrix}0\\ \vdots\\ 0\\ \beta_{i-2}\\ 1\\ 0\\ \vdots\\ 0\end{bmatrix}\begin{matrix}(i-2)\\(i-1)\end{matrix} = r_i,$$

所以 $r_i = a_i$，$i=2,3,\cdots,n$.

又因为

$$\underset{(i,i)}{\underset{\downarrow}{b_i}} = L_{i*}U_{*i} = (0,\cdots,0,\ \underset{(i-1)}{\underset{\downarrow}{r_i}},\ \underset{(i)}{\underset{\downarrow}{\alpha_i}},0,\cdots,0)\begin{bmatrix}0\\ \vdots\\ 0\\ \beta_{i-1}\\ 1\\ 0\\ \vdots\\ 0\end{bmatrix}\begin{matrix}(i-1)\\(i)\end{matrix}$$

$$= r_i\beta_{i-1} + \alpha_i = \alpha_i + a_i\beta_{i-1},$$

所以 $\alpha_i = b_i - a_i\beta_{i-1}$，$i=2,3,\cdots,n$.

因为

$$b_1 = L_{1*}U_{*1} = (\alpha_1,0,\cdots,0)\begin{bmatrix}1\\ 0\\ \vdots\\ 0\end{bmatrix} = \alpha_1,$$

所以 $\alpha_1 = b_1$.

又因为

$$\underset{(i,i+1)}{\underset{\downarrow}{c_i}} = L_{i*}U_{*i+1} = \alpha_i\beta_i,$$

所以 $\beta_i = c_i/\alpha_i$，$i=1,2,\cdots,n-1$.

归纳上述公式得式(3－13)

$$\begin{aligned}
&r_i = a_i, && i=2,3,\cdots,n;\\
&\begin{cases}\alpha_1 = b_1,\\ \alpha_i = b_i - a_i\beta_{i-1}, & i=2,3,\cdots,n;\end{cases}\\
&\beta_i = c_i/\alpha_i, && i=1,\cdots,n-1.
\end{aligned}$$

(2)公式(3－14)的推导.

设下三角方程组 $Ly=f$，即

$$\begin{bmatrix} \alpha_1 & & & & & & \\ a_2 & \alpha_2 & & & & & \\ & \ddots & \ddots & & & & \\ & & a_i & \alpha_i & & & \\ & & & \ddots & \ddots & & \\ & & & & a_{n-1} & \alpha_{n-1} & \\ & & & & & a_n & \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} f_1 \\ \vdots \\ f_i \\ \vdots \\ f_n \end{bmatrix}.$$

由 $L_{1*}y=f_1$ 得 $\alpha_1 y_1=f_1$，即 $y_1=f_1/\alpha_1$.

又由 $L_{i*}y=f_i$，即

$$(0,\cdots,0,a_i,\alpha_i,0,\cdots,0)\begin{bmatrix} y_1 \\ \vdots \\ y_{i-1} \\ y_i \\ \vdots \\ y_n \end{bmatrix}=f_i$$

得
$$a_i y_{i-1}+\alpha_i y_i=f_i.$$
所以
$$y_i=(f_i-a_i y_{i-1})/\alpha_i.$$

综上，得到式(3－14)

$$\begin{cases} y_1=f_1/b_1, \\ y_i=(f_i-a_i y_{i-1})/\alpha_i, \quad i=2,3,\cdots,n. \end{cases}$$

(3)公式(3－15)的推导.

设上三角方程组 $Ux=y$，即

$$\begin{bmatrix} 1 & \beta_1 & & & & & \\ & 1 & \beta_2 & & & & \\ & & \ddots & \ddots & & & \\ & & & 1 & \beta_i & & \\ & & & & \ddots & \ddots & \\ & & & & & 1 & \beta_{i-1} \\ & & & & & & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_i \\ x_{i+1} \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_i \\ y_{i+1} \\ \vdots \\ y_n \end{bmatrix}.$$

由 $U_{n*}x=y_n$，即

$$(0,0,\cdots,0,1)\begin{bmatrix} x_1 \\ \vdots \\ x_i \\ x_{i+1} \\ \vdots \\ x_n \end{bmatrix}=y_n,$$

得 $x_n=y_n$.

又由 $U_{i*}x=y_i$，即

$$(0,\cdots,\underset{(i)}{1},\underset{(i+1)}{\beta_i},0,\cdots,0)\begin{bmatrix}x_1\\ \vdots\\ x_i\\ x_{i+1}\\ \vdots\\ x_n\end{bmatrix}=y_i,$$

得 $$x_i+\beta_i x_{i+1}=y_i,$$

所以 $$x_i=y_i-\beta_i x_{i+1}.$$

综上可得式(3−15)

$$\begin{cases}x_n=y_n,\\ x_i=y_i-\beta_i x_{i+1}, \quad i=n-1,n-2,\cdots,1.\end{cases}$$

习题 3

*1. 用 Gauss 消元法求解下列方程组 $\boldsymbol{Ax}=\boldsymbol{b}$.

(1)$\boldsymbol{A}=\begin{bmatrix}1&1&-1\\1&2&-2\\-2&1&1\end{bmatrix}$, $\boldsymbol{b}=\begin{bmatrix}1\\0\\1\end{bmatrix}$;

(2)$\boldsymbol{A}=\begin{bmatrix}4&3&2&1\\3&4&3&2\\2&3&4&3\\1&2&3&4\end{bmatrix}$, $\boldsymbol{b}=\begin{bmatrix}1\\1\\-1\\-1\end{bmatrix}$.

*2. 用列主元消元法解下列方程组 $\boldsymbol{Ax}=\boldsymbol{b}$.

(1)$\boldsymbol{A}=\begin{bmatrix}-3&2&6\\10&-7&0\\5&-1&5\end{bmatrix}$, $\boldsymbol{b}=\begin{bmatrix}4\\7\\6\end{bmatrix}$;

(2)$\boldsymbol{A}=\begin{bmatrix}0&2&0&1\\2&2&3&2\\4&-3&0&1\\6&1&-6&-5\end{bmatrix}$, $\boldsymbol{b}=\begin{bmatrix}0\\-2\\-7\\6\end{bmatrix}$.

*3. 用 Gauss−Jordan 消元法求 $\begin{bmatrix}1&1&-1\\2&1&0\\1&-1&0\end{bmatrix}^{-1}$.

*4. 用直接 $\boldsymbol{LU}$ 分解法求第 1 题中两个矩阵的 $\boldsymbol{LU}$ 分解,并求解这两个方程组.

*5. 用追赶法解三对角方程组 $\boldsymbol{Ax}=\boldsymbol{b}$,其中:

$$
\boldsymbol{A}=\begin{bmatrix} 2 & -1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix}, \quad \boldsymbol{b}=\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.
$$

6. 由 $\boldsymbol{L}=\boldsymbol{F}_1^{-1}\boldsymbol{F}_2^{-1}\cdots\boldsymbol{F}_{n-1}^{-1}$，证明：

$$
\boldsymbol{L}=\begin{bmatrix} 1 & & & & & \\ l_{21} & 1 & & & & \\ l_{31} & l_{32} & 1 & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ \vdots & \vdots & \vdots & & 1 & \\ l_{n1} & l_{n2} & l_{n3} & \cdots & l_{n,n-1} & 1 \end{bmatrix}.
$$

7. 设 $\boldsymbol{A}$ 是对称正定矩阵，且经过 Gauss 消元法第一步后，$\boldsymbol{A}$ 约化为

$$
\begin{pmatrix} a_{11} & \boldsymbol{a}_1^{\mathrm{T}} \\ \boldsymbol{0} & \boldsymbol{A}_2 \end{pmatrix}.
$$

其中，$\boldsymbol{A}=(a_{ij})_{n\times n}$，$\boldsymbol{A}_2=(a_{ij}^{(2)})_{(n-1)\times(n-1)}$．证明：

(1) $\boldsymbol{A}$ 的对角元素 $a_{ii}>0, i=1,2,\cdots,n$；

(2) $\boldsymbol{A}_2$ 是对称正定矩阵；

(3) $a_{ii}^{(2)}\leqslant a_{ii}, i=2,3,\cdots,n$.

8. 设 $\boldsymbol{U}\boldsymbol{x}=\boldsymbol{b}$，其中 $\boldsymbol{U}$ 为三角矩阵.

(1)就 $\boldsymbol{U}$ 为上三角及下三角矩阵推导一般的求解公式，并写出算法.

(2)设 $\boldsymbol{U}$ 为非奇异阵，试推导求 $\boldsymbol{U}^{-1}$ 的计算公式.

注　打 * 号的习题在本书附录Ⅱ作了详解.

第 4 章　线性方程组的迭代解法

解线性方程组的直接解法通常适用于阶数不太高的线性方程组，而对于大型稀疏线性方程组则宜于采用迭代法。迭代法可以保持系数矩阵稀疏的优点，从而节约大量的存储单元和计算工作量。

用迭代法解线性方程组，就是构造一个无限的向量序列，使它的极限成为线性方程组的解向量。因此我们首先介绍 n 维实欧几里得(Euclid)空间 R^n 上的向量范数以及相应的矩阵范数。

4.1　向量范数与矩阵范数

4.1.1　向量范数

定义 4.1(p 范数)　设向量 $x\in R^n$，称 $\|x\|_p=(\sum\limits_{i=1}^{n}|x_i|^p)^{1/p}\ (p\geqslant 1)$ 为 x 在 R^n 上的 p 范数.

当 $p=1$，称 $\|x\|_1=\sum\limits_{i=1}^{n}|x_i|$ 为 x 的 1 范数；

当 $p=2$，称 $\|x\|_2=(\sum\limits_{i=1}^{n}|x_i|^2)^{1/2}=(\sum\limits_{i=1}^{n}x_i^2)^{1/2}=(x,x)^{1/2}$ 为 x 的 2 范数，又称为 Euclid 范数；

当 $p=\infty$，称 $\|x\|_\infty=\lim\limits_{p\to\infty}\|x\|_p$ 为 x 的无穷大范数.

容易证明

$$\|x\|_\infty=\lim_{p\to\infty}\|x\|_p=\max_{1\leqslant i\leqslant n}|x_i|. \tag{4-1}$$

因为

$$\max_{1\leqslant i\leqslant n}|x_i|^p\leqslant\sum_{i=1}^{n}|x_i|^p\leqslant n\cdot\max_{1\leqslant i\leqslant n}|x_i|^p,$$

即

$$\max_{1\leqslant i\leqslant n}|x_i|\leqslant(\sum_{i=1}^{n}|x_i|^p)^{1/p}\leqslant n^{1/p}\cdot\max_{1\leqslant i\leqslant n}|x_i|,$$

$$1\leqslant\frac{\|x\|_p}{\max\limits_{1\leqslant i\leqslant n}|x_i|}\leqslant n^{1/p}.$$

令 $p\to\infty$ 得

$$\frac{\lim\limits_{p\to\infty}\|x\|_p}{\max\limits_{1\leqslant i\leqslant n}|x_i|}=1,$$

即

$$\lim_{p\to\infty}\|x\|_p=\max_{1\leqslant i\leqslant n}|x_i|.$$

由此证得式(4-1)成立.

当 $x\in R^n$ 时，x 的三种 p 范数在图 4-1 中有直观的几何意义. $\|x\|_2$ 代表直角三角

形斜边的长，$\|x\|_1$ 代表两直角边边长之和，$\|x\|_\infty$ 代表最大直角边的边长. 由此可见，向量的长度有多种度量方式.

例 1 设 $x=(1,0.5,0,-0.3)^T$，求 $\|x\|_1$，$\|x\|_2$，$\|x\|_\infty$.

解 $\|x\|_1=1+0.5+0+0.3=1.8$，

$\|x\|_2=\sqrt{1^2+0.5^2+0.3^2}=1.157\,6$，

$\|x\|_\infty=1$.

可以证明，$x\in R^n$ 的三种 p 范数满足不等式

$$\|x\|_1\geqslant\|x\|_2\geqslant\|x\|_\infty.$$

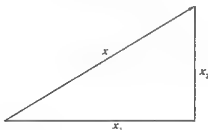


图 4－1

定义 4.2（向量范数的一般定义） 设 $x\in R^n$，$\|x\|$ 是定义在 R^n 上的实值函数，它满足：

(1) $\|x\|\geqslant 0$，当且仅当 $x=0$ 时，$\|x\|=0$；

(2) $\|kx\|=|k|\,\|x\|$，k 是任意实数；

(3) $\|x+y\|\leqslant\|x\|+\|y\|$，其中 $y\in R^n$.

我们称 $\|x\|$ 为 x 在 R^n 上的向量范数. 称(3)为三角不等式. 而更一般的三角不等式为

$$\|x\|-\|y\|\leqslant\|x\pm y\|\leqslant\|x\|+\|y\|.$$

4.1.2 向量序列的收敛性

定义 4.3 设 $\{x^{(k)}\}$ 是 R^n 上的向量序列，$x^*\in R^n$，$\lim\limits_{k\to\infty}x^{(k)}=x^*$，当且仅当 $\lim\limits_{k\to\infty}x_i^{(k)}=x_i^*$ $(i=1,2,\cdots,n)$，其中

$$x^{(k)}=(x_1^{(k)},x_2^{(k)},\cdots,x_n^{(k)})^T,$$

$$x^*=(x_1^*,x_2^*,\cdots,x_n^*)^T.$$

上述定义表明，一个向量序列的收敛等价于 n 个分量序列（实数数列）的收敛. 显然按定义 4.3 检查向量序列的收敛性是相当麻烦的，为此引入以下判别定理.

定理 4.1（向量序列收敛的充要条件） 设 $\{x^{(k)}\}$，$x^{(k)}\in R^n$，$x^*\in R^n$，$\lim\limits_{k\to\infty}x^{(k)}=x^*$ 的充要条件是

$$\lim_{k\to\infty}\|x^{(k)}-x^*\|=0. \tag{4-2}$$

定理 4.1 的证明从略. 式(4－2)表明，一个向量序列的收敛等价于一个实数数列的收敛.

4.1.3 矩阵范数

定义 4.4(矩阵的 $p=1,2,\infty$范数) 设 $A=(a_{ij})_{n\times n}$,则称

(1) $\|A\|_\infty=\max\limits_{1\leqslant i\leqslant n}\sum\limits_{j=1}^{n}|a_{ij}|$ 为 A 的无穷大范数,又称为 A 的行模;

(2) $\|A\|_1=\max\limits_{1\leqslant j\leqslant n}\sum\limits_{i=1}^{n}|a_{ij}|$ 为 A 的 1 范数,又称为 A 的列模;

(3) $\|A\|_2=(\max|\lambda_{A^TA}|)^{1/2}$为 A 的 2 范数,又称为 A 的谱模.

例 2 设
$$A=\begin{pmatrix}1 & -2\\ -3 & 4\end{pmatrix},\quad B=\begin{bmatrix}-2 & 1 & 0\\ 1 & -2 & 1\\ 0 & 1 & -2\end{bmatrix},$$
求 $\|A\|_{p=1,2,\infty}$, $\|B\|_{p=1,2,\infty}$.

解 $\|A\|_1=\max(4,6)=6,\quad \|A\|_\infty=\max(3,7)=7;$

$\|B\|_1=\max(3,4,3)=4,\quad \|B\|_\infty=\max(3,4,3)=4.$

下面分别求 $\|A\|_2$, $\|B\|_2$:

因为
$$A^TA=\begin{pmatrix}10 & -14\\ -14 & 20\end{pmatrix},$$
则
$$\det(\lambda I-A^TA)=\begin{vmatrix}\lambda-10 & 14\\ 14 & \lambda-20\end{vmatrix}=\lambda^2-30\lambda+4.$$
令 $\det(\lambda I-A^TA)=0$,即
$$\lambda^2-30\lambda+4=0,$$
解得
$$\lambda_{1,2}=15\pm\sqrt{221}.$$
所以
$$\|A\|_2=(15+\sqrt{221})^{1/2}=5.465.$$
又因为
$$B^TB=\begin{bmatrix}5 & -4 & 1\\ -4 & 6 & -4\\ 1 & -4 & 5\end{bmatrix},$$
则
$$\begin{aligned}\det(\lambda I-B^TB)&=\begin{vmatrix}\lambda-5 & 4 & -1\\ 4 & \lambda-6 & 4\\ -1 & 4 & \lambda-5\end{vmatrix}\\&=\lambda^3-16\lambda^2+52\lambda-16\\&=(\lambda^3-12\lambda^2+4\lambda)-(4\lambda^2-48\lambda+16)\\&=\lambda(\lambda^2-12\lambda+4)-4(\lambda^2-12\lambda+4)\\&=(\lambda^2-12\lambda+4)(\lambda-4).\end{aligned}$$
令 $\det(\lambda I-B^TB)=0$,即
$$(\lambda^2-12\lambda+4)(\lambda-4)=0,$$
解得
$$\lambda_1=4,\quad \lambda_{2,3}=6\pm4\sqrt{2}.$$
所以
$$\|B\|_2=(6+4\sqrt{2})^{1/2}=3.414\ 2.$$

定义 4.5(矩阵范数的一般定义)　设 $A=(a_{ij})_{n\times n}$，$x\in R^n$，则称

$$\|A\|=\max_{x\neq 0}\frac{\|Ax\|}{\|x\|}=\max_{\|x\|=1}\|Ax\|$$

为矩阵 A 的范数.

定义 4.5 可以理解为 A 作为线性变换作用于不同的 x 后，能将 x 的范数放大的最大倍数. 这样定义的矩阵范数有如下性质：

(1)非负性. $\|A\|\geqslant 0$，$\|A\|=0$ 当且仅当 $A=\boldsymbol{\theta}$ 时成立.

(2)齐次性. $\|kA\|=|k|\,\|A\|$，k 为一个实数.

(3)三角不等式. 设 A,B 都是 $n\times n$ 阶方阵，则有 $\|A+B\|\leqslant\|A\|+\|B\|$.

(4)相容性. 设 A,B 为同阶方阵，则有 $\|AB\|\leqslant\|A\|\cdot\|B\|$.

(5)相容性条件. 设 A 为 $n\times n$ 阶方阵，$x\in R^n$，则有 $\|Ax\|\leqslant\|A\|\cdot\|x\|$.

4.1.4　矩阵的特征值上界

定义 4.6(谱半径)　设 $A=(a_{ij})_{n\times n}$，$\lambda_i(i=1,2,\cdots,n)$ 为 A 的 n 个特征值，则称

$$\rho(A)=\max_{1\leqslant i\leqslant n}|\lambda_i|$$

为矩阵 A 的谱半径.

例 3　求例 2 中矩阵 A,B 的谱半径.

解　因为
$$A=\begin{pmatrix}1&-2\\-3&4\end{pmatrix},$$
而
$$\det(\lambda I-A)=\begin{vmatrix}\lambda-1&2\\3&\lambda-4\end{vmatrix}=\lambda^2-5\lambda-2=0,$$
解得
$$\lambda_{1,2}=\frac{5\pm\sqrt{33}}{2}.$$
所以
$$\rho(A)=\frac{5+\sqrt{33}}{2}=5.372.$$

又因为
$$B=\begin{bmatrix}-2&1&0\\1&-2&1\\0&1&-2\end{bmatrix},$$
$$\det(\lambda I-B)=\begin{vmatrix}\lambda+2&-1&0\\-1&\lambda+2&-1\\0&-1&\lambda+2\end{vmatrix}=(\lambda+2)(\lambda^2+4\lambda+2)=0,$$
解得
$$\lambda_1=-2,\quad \lambda_{2,3}=-2\pm\sqrt{2}.$$
所以
$$\rho(B)=2+\sqrt{2}=3.414\ 2.$$

例 4　设 $C=\begin{bmatrix}0&3/8&-1/4\\-4/11&0&1/11\\-1/2&-1/4&0\end{bmatrix}$，求 $\rho(C)$.

解 因为

$$\det(\lambda I-C)=\begin{vmatrix}\lambda & -3/8 & 1/4\\ 4/11 & \lambda & -1/11\\ 1/2 & 1/4 & \lambda\end{vmatrix}$$

$$=\lambda^3+0.034\,09\lambda+0.397\,7,$$

令

$$f(\lambda)=\lambda^3+0.034\,09\lambda+0.397\,7=0,$$

用 Newton 迭代法

$$\lambda_{k+1}=\lambda_k-f(\lambda_k)/f'(\lambda_k)$$

求得 $f(\lambda)=0$ 的一个实根 $\lambda_1=-0.308\,2$，再用 Newton 迭代法求该方程余下的两个复根

$$\lambda_{2,3}=0.154\,1\pm0.324\,5\mathrm{i}.$$

所以

$$\rho(C)=\max_{1\leq i\leq 3}|\lambda_i|$$

$$=\max(0.308\,2,0.359\,2,0.359\,2)$$

$$=0.359\,2.$$

定理 4.2(谱半径上界) 设 $A=(a_{ij})_{n\times n}$，则 $\rho(A)\leqslant\|A\|$.

证 设 λ_i 是 A 的任一特征值，x_i 为对应的特征向量，据特征值定义

$$Ax_i=\lambda_i x_i,$$

两端取范数，有

$$\|Ax_i\|=\|\lambda_i x_i\|=|\lambda_i|\cdot\|x_i\|. \tag{4-3}$$

据矩阵范数相容性条件有

$$\|Ax_i\|\leqslant\|A\|\cdot\|x_i\|, \tag{4-4}$$

比较(4-3),(4-4)两式得

$$|\lambda_i|\,\|x_i\|\leqslant\|A\|\cdot\|x_i\|. \tag{4-5}$$

因为 $x_i\neq 0$，所以 $\|x_i\|>0$，故有

$$|\lambda_i|\leqslant\|A\|,\quad 即\ \rho(A)\leqslant\|A\|.$$

还应指出的是，对于实对称矩阵 A，有 $\rho(A)=\|A\|_2$ 成立.

归纳例 2,例 3 的计算结果，我们有

$$A=\begin{pmatrix}1 & -2\\ -3 & 4\end{pmatrix},\quad \|A\|_1=6,\quad \|A\|_\infty=7,\quad \|A\|_2=5.465,\quad \rho(A)=5.372;$$

$$\rho(A)<\|A\|_1,\|A\|_2,\|A\|_\infty.$$

$$B=\begin{bmatrix}-2 & 1 & 0\\ 1 & -2 & 1\\ 0 & 1 & -2\end{bmatrix},\quad \|B\|_1=4,\quad \|B\|_\infty=4,\quad \|B\|_2=3.414\,2,\quad \rho(B)=3.414\,2;$$

$$\rho(B)<\|B\|_1,\|B\|_\infty,\quad 但\ \rho(B)=\|B\|_2.$$

4.1.5 矩阵序列的收敛性

设 $B=(b_{ij})_{n\times n}$，称 $\{B^k\}_{k=1,2,\cdots}$ 为乘幂矩阵序列，如 $B,B^2,B^3,\cdots,B^k,\cdots$

定理 4.3 设 $\{B^k\}$ 为乘幂矩阵序列，则 $B^k\to\theta(k\to\infty)$ 的充分必要条件是 $\rho(B)<1$(证明从略).

例如，在例 3 中，因 $\rho(A)>1,\rho(B)>1$，所以

$$A^k \to \theta \quad (k \to \infty),$$
$$B^k \to \theta \quad (k \to \infty).$$

而在例 4 中，$\rho(C)=0.3592<1$，所以

$$C^k \to \theta \quad (k \to \infty).$$

4.2 迭代法

4.2.1 问题的提出

在第 2 章中我们讨论了解非线性方程的迭代法：对方程 $f(x)=0$ 作恒等变形，即

$$f(x)=0 \xLeftrightarrow{\text{令} f(x)=x-\varphi(x)} x-\varphi(x)=0,$$

由此构成迭代公式

$$\begin{cases} x_{k+1}=\varphi(x_k), & k=0,1,\cdots \\ x_0(\text{给定}). \end{cases} \tag{4-6}$$

本章我们采用同样的思路研究解线性方程组的迭代法.

对线性方程组作恒等变形

$$Ax=b \xLeftrightarrow[\det(N)\neq 0]{\text{令} A=N-P} x=Bx+d,$$

其中

$$B=N^{-1}P, \quad d=N^{-1}b,$$

构成迭代公式

$$\begin{cases} x^{(k+1)}=Bx^{(k)}+d, & k=0,1,\cdots \\ x^{(0)}(\text{给定}). \end{cases} \tag{4-7}$$

由式(4－7)产生迭代点列 $\{x^{(k)}\}_{k=0,1,\cdots}$. 若 $\lim\limits_{k\to\infty} x^{(k)}=x^*$，对式(4－7)两端取极限，有

$$x^*=Bx^*+d,$$

则我们称 x^* 是方程组 $x=Bx+d$ 的不动点，即是方程组 $Ax=b$ 的解.

本节主要讨论两种迭代公式，即雅可比(Jacobi)迭代式与高斯－塞德尔(Gauss－Seidel)迭代式，并讨论它们的收敛条件.

4.2.2 雅可比(Jacobi)迭代法

4.2.2.1 Jacobi 迭代式的矩阵形式

设 $Ax=b$，并令

$$A=L+D+U=D+(L+U),$$

其中

$$L=\begin{bmatrix} 0 & & & & \\ a_{21} & 0 & & & \\ a_{31} & a_{32} & 0 & & \\ \vdots & \vdots & \ddots & \ddots & \\ a_{n1} & a_{n2} & \cdots & a_{n,n-1} & 0 \end{bmatrix},$$

$$D=\begin{bmatrix} a_{11} & & & \\ & a_{22} & & \\ & & \ddots & \\ & & & a_{nn} \end{bmatrix},$$

且 $a_{ii}\neq 0, i=1,2,\cdots,n$，即

$$\det(D)\neq 0,$$

$$U=\begin{bmatrix} 0 & a_{12} & a_{13} & \cdots & a_{1n} \\ & 0 & a_{23} & \cdots & a_{2n} \\ & & \ddots & & \vdots \\ & & & \ddots & \\ & & & & 0 \end{bmatrix}.$$

于是原方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 变成

$$[\boldsymbol{D}+(\boldsymbol{L}+\boldsymbol{U})]\boldsymbol{x}=\boldsymbol{b},$$

即

$$\boldsymbol{Dx}=-(\boldsymbol{L}+\boldsymbol{U})\boldsymbol{x}+\boldsymbol{b}.$$

因为 $\det(\boldsymbol{D})\neq 0$，所以有

$$\boldsymbol{x}=-\boldsymbol{D}^{-1}(\boldsymbol{L}+\boldsymbol{U})\boldsymbol{x}+\boldsymbol{D}^{-1}\boldsymbol{b}.$$

令

$$\boldsymbol{B}_J=-\boldsymbol{D}^{-1}(\boldsymbol{L}+\boldsymbol{U}),\quad \boldsymbol{d}_J=\boldsymbol{D}^{-1}\boldsymbol{b},$$

得

$$\begin{cases}\boldsymbol{x}^{(k+1)}=\boldsymbol{B}_J\,\boldsymbol{x}^{(k)}+\boldsymbol{d}_J,\\ \boldsymbol{B}_J=-\boldsymbol{D}^{-1}(\boldsymbol{L}+\boldsymbol{U}),\quad \boldsymbol{d}_J=\boldsymbol{D}^{-1}\boldsymbol{b}.\end{cases}\tag{4-8}$$

我们称式(4－8)为 Jacobi 迭代式(矩阵形式).

4.2.2.2　*Jacobi 迭代式分量形式*

由矩阵形式 $\boldsymbol{x}^{(k+1)}=\boldsymbol{B}_J\boldsymbol{x}^{(k)}+\boldsymbol{d}_J$ 可得下列分量形式：

$$x_i^{(k+1)}=(\boldsymbol{B}_J\boldsymbol{x}^{(k)})_i+(\boldsymbol{d}_J)_i=(\boldsymbol{B}_J)_{i*}\boldsymbol{x}^{(k)}+(\boldsymbol{d}_J)_i.$$

式中

$$(\boldsymbol{B}_J)_{i*}=\left(-\frac{a_{i1}}{a_{ii}},\cdots,-\frac{a_{i,i-1}}{a_{ii}},0,-\frac{a_{i,i+1}}{a_{ii}},\cdots,-\frac{a_{in}}{a_{ii}}\right),$$

$$\boldsymbol{d}_J=\left(\frac{b_1}{a_{11}},\cdots,\frac{b_i}{a_{ii}},\cdots,\frac{b_n}{a_{nn}}\right)^{\mathrm{T}},$$

$$(\boldsymbol{d}_J)_i=\frac{b_i}{a_{ii}},\quad i=1,2,\cdots,n.$$

所以

$$x_i^{(k+1)}=(b_i-\sum_{\substack{j=1\\ j\neq i}}^{n}a_{ij}\,x_j^{(k)})/a_{ii},\quad i=1,2,\cdots,n.\tag{4-9}$$

亦即

$$\begin{cases}x_1^{(k+1)}=(b_1-0-a_{12}x_2^{(k)}-a_{13}x_3^{(k)},\cdots,-a_{1n}x_n^{(k)})/a_{11},\\ x_2^{(k+1)}=(b_2-a_{21}x_1^{(k)}-0-a_{23}x_3^{(k)},\cdots,-a_{2n}x_n^{(k)})/a_{22},\\ \quad\vdots\\ x_n^{(k+1)}=(b_n\quad a_{n1}x_1^{(k)}\quad a_{n2}x_2^{(k)}\quad a_{n3}x_3^{(k)},\cdots,-a_{n,n-1}x_{n-1}^{(k)}-0)/a_{nn}.\end{cases}\tag{4-10}$$

式(4－9)为 Jacobi 迭代式分量形式的紧凑格式，而式(4　10)为 Jacobi 迭代式分量形式的展开格式.

4.2.2.3 计算表格

例5 用 Jacobi 迭代法解线性方程组 $\boldsymbol{Ax}=\boldsymbol{b}$，其中

$$\boldsymbol{A}=\begin{bmatrix}8 & -3 & 2\\ 4 & 11 & -1\\ 6 & 3 & 12\end{bmatrix},\quad \boldsymbol{b}=(20,33,36)^{\mathrm{T}}.$$

解 (1)建立迭代公式

$$\boldsymbol{x}^{(k+1)}=\boldsymbol{B}_{\mathrm{J}}\boldsymbol{x}^{(k)}+\boldsymbol{d}_{\mathrm{J}},$$

其中
$$\boldsymbol{B}_{\mathrm{J}}=-\boldsymbol{D}^{-1}(\boldsymbol{L}+\boldsymbol{U})=\begin{bmatrix}0 & 3/8 & -1/4\\ -4/11 & 0 & 1/11\\ -1/2 & -1/4 & 0\end{bmatrix},$$

$$\boldsymbol{d}_{\mathrm{J}}=\boldsymbol{D}^{-1}\boldsymbol{b}=(2.5,3,3)^{\mathrm{T}}.$$

(2)列表计算(见表4－1，保留6位小数)：

表 4－1

$\boldsymbol{x}^{(k)}$	$\boldsymbol{B}_{\mathrm{J}}=-\boldsymbol{D}^{-1}(\boldsymbol{L}+\boldsymbol{U})$			$\boldsymbol{d}_{\mathrm{J}}=\boldsymbol{D}^{-1}\boldsymbol{b}$
	0	3/8	−1/4	2.5
	−4/11	0	1/11	3
	−1/2	−1/4	0	3
$\boldsymbol{x}^{(0)}$	0	0	0	
$\boldsymbol{x}^{(1)}$	2.5	3	3	
$\boldsymbol{x}^{(2)}$	2.875	2.364	1	
$\boldsymbol{x}^{(3)}$	3.136 5	2.045 5	0.971 5	
$\boldsymbol{x}^{(4)}$	3.024 18	1.947 77	0.920 38	
$\boldsymbol{x}^{(5)}$	3.000 318	1.983 965	1.000 963	
$\boldsymbol{x}^{(6)}$	2.993 746	1.999 971	1.003 849	
$\boldsymbol{x}^{(7)}$	2.999 026	2.002 624	1.003 134	
$\boldsymbol{x}^{(8)}$	3.000 201	2.000 639	0.999 831	
$\boldsymbol{x}^{(9)}$	3.000 282	1.999 911	0.999 740	
$\boldsymbol{x}^{(10)}$	3.000 032	1.999 838	0.999 813	

原方程组的精确解 $\boldsymbol{x}^{*}=(3,2,1)^{\mathrm{T}}$，由表4－1可以看出，迭代向量 $\boldsymbol{x}^{(k)}$ 逐步逼近 $\boldsymbol{x}^{*}$，且 $\|\boldsymbol{x}^{(10)}-\boldsymbol{x}^{*}\|_{\infty}=0.000\,187<0.5\times10^{-3}$.

(3)终止法则：

$$\begin{aligned}\|\boldsymbol{x}^{(10)}-\boldsymbol{x}^{(9)}\|_{\infty}&=\max_{1\leqslant i\leqslant 3}|x_i^{(10)}-x_i^{(9)}|\\&=\max(0.000\,25,0.000\,073,0.000\,073)\\&=0.000\,25<0.5\times10^{-3},\end{aligned}$$

所以
$$\boldsymbol{x}^{*}\approx\boldsymbol{x}^{(10)}=(3.000\,032,1.999\,838,0.999\,813)^{\mathrm{T}}\approx(3,2,1)^{\mathrm{T}}.$$

4.2.2.4 算法设计

(1)数据存储方式.

$A[1:n,1:n]$,二维数组,存放系数矩阵的元素;

$b[1:n]$,一维数组,存放右端向量的分量;

$x[1:n]$,一维数组,存放每次迭代的后继向量 $x^{(k+1)}$;

$y[1:n]$,一维数组,存放每次迭代的当前向量 $x^{(k)}$.

(2)迭代公式.

$$x_i=(b_i-\sum_{\substack{j=1\\j\neq i}}^{n}a_{ij}y_j)/a_{ii},\quad i=1,2,\cdots,n.$$

计算 $x_i(i=1,2,\cdots,n)$时,使用同一组初值 $y_i(i=1,2,\cdots,n)$,如下所示:

$$\begin{bmatrix}x_1\\x_2\\\vdots\\x_n\end{bmatrix}\begin{matrix}\leftarrow & y_1 & y_2 & \cdots & y_n\\\leftarrow & y_1 & y_2 & \cdots & y_n\\ & \vdots & \vdots & & \vdots\\\leftarrow & y_1 & y_2 & \cdots & y_n\end{matrix}$$

一次迭代完毕,$x_i\to y_i(i=1,2,\cdots,n)$,再进入下一次迭代.

(3)终止法则.

检查 $\|x-y\|_\infty=\max\limits_{1\le i\le n}|x_i-y_i|<\varepsilon$?

是,$x\Rightarrow x^*$, stop;

否,$x_i\Rightarrow y_i(i=1,2,\cdots,n)$,继续执行下一次迭代.

(4)算法.

Algo. 4.1

1° 读入初始数据 A,b,ε.

2° 初始向量赋值,即 $0\to y_i(i=1,2,\cdots,n)$.

3° $1\to i$.

4° 计算$(b_i-\sum\limits_{\substack{j=1\\j\neq i}}^{n}a_{ij}y_j)/a_{ii}\to x_i$.

5° 检查:$i=n$?

是,转 6°;

否,$i+1\to i$,转 4°.

6° $1\to k$.

7° 检查:$|x_k-y_k|<\varepsilon$?

是,转 8°;

否,$x_j\to y_j(j=1,2,\cdots,n)$,转 3°.

8° 检查:$k=n$?

是,输出 $x_j(j=1,2,\cdots,n)$, stop;

否,$k+1\to k$,转 7°.

4.2.3 高斯－塞德尔(Gauss－Seidel)迭代法

4.2.3.1 问题的提出

Jacobi 迭代法的特征是同步迭代，下面以第一次迭代为例来加以说明.

给定初值 $x^{(0)}=\begin{bmatrix}x_1^{(0)}\\x_2^{(0)}\\\vdots\\x_n^{(0)}\end{bmatrix}$，计算 $x^{(1)}$：

$$x^{(1)}=\begin{bmatrix}x_1^{(1)}\\x_2^{(1)}\\\vdots\\x_n^{(1)}\end{bmatrix}\begin{matrix}\leftarrow & x_1^{(0)} & x_2^{(0)} & \cdots & x_n^{(0)}\\\leftarrow & x_1^{(0)} & x_2^{(0)} & \cdots & x_n^{(0)}\\ & \vdots & \vdots & & \vdots\\\leftarrow & x_1^{(0)} & x_2^{(0)} & \cdots & x_n^{(0)}\end{matrix}$$

所谓同步是指计算 $x^{(1)}$ 的每一个分量都是用 $x^{(0)}$ 的 n 个分量 $x_1^{(0)},x_2^{(0)},\cdots,x_n^{(0)}$（同一组初值），计算公式用式(4－9)或式(4－10).

保持同步的优点是计算简单，缺点是计算速度慢. Gauss－Seidel 迭代法采用异步迭代的方式，即

$$x^{(1)}=\begin{bmatrix}x_1^{(1)}\\x_2^{(1)}\\\vdots\\x_n^{(1)}\end{bmatrix}\begin{matrix}\leftarrow & x_1^{(0)} & x_2^{(0)} & \cdots & x_n^{(0)}\\\leftarrow & x_1^{(1)} & x_2^{(0)} & \cdots & x_n^{(0)}\\ & \vdots & \vdots & \cdots & \vdots\\\leftarrow & x_1^{(1)} & x_2^{(1)} & \cdots & x_n^{(0)}\end{matrix}$$

这种异步迭代的特点是用新计算出来的分量逐个替换旧的分量.

4.2.3.2 Gauss－Seidel 迭代公式(分量形式)

对 Jacobi 迭代公式的分量形式，即式(4－9)作如下变形：

$$\begin{aligned}x_i^{(k+1)}&=\Big(b_i-\sum_{\substack{j=1\\j\neq i}}^{n}a_{ij}x_j^{(k)}\Big)/a_{ii}\\&=\Big(b_i-\sum_{j=1}^{i-1}a_{ij}x_j^{(k)}-\sum_{j=i+1}^{n}a_{ij}x_j^{(k)}\Big)/a_{ii},\quad i=1,2,\cdots,n,\end{aligned}$$

其中，若将 x^{k+1} 的分量逐个替换 $x^{(k)}$ 的相应分量，便得

$$x_i^{(k+1)}=\Big(b_i-\sum_{j=1}^{i-1}a_{ij}x_j^{(k+1)}-\sum_{j=i+1}^{n}a_{ij}x_j^{(k)}\Big)/a_{ii},\quad i=1,2,\cdots,n.\tag{4－11}$$

称式(4－11)为 Gauss－Seidel 迭代式的分量形式.

4.2.3.3 Gauss－Seidel 迭代公式(矩阵形式)

在 Jacobi 迭代公式的矩阵形式，即式(4－8)中

$$x^{(k+1)}=B_Jx^{(k)}+d_J,$$

$$B_J=-D^{-1}(L+U)=-D^{-1}L-D^{-1}U,$$

$$d_J=D^{-1}b.$$

若令

$$B_{J1}=-D^{-1}L,\quad B_{J2}=-D^{-1}U,$$

则得到 Gauss－Seidel 迭代公式的矩阵形式

$$x^{(k+1)}=B_{J1}x^{(k+1)}+B_{J2}x^{(k)}+d_J. \tag{4-12}$$

4.2.3.4　计算表格

用 Gauss－Seidel 迭代式(4－12)求解例 5 中的方程组，并列表计算(结果见表 4－2，保留 6 位小数)。

表 4－2

$x^{(k)}$	$B_J=-D^{-1}(L+U)$			$d_J=D^{-1}b$
	0	3/8	−1/4	2.5
	−4/11	0	1/11	3
	−1/2	−1/4	0	3
$x^{(0)}$	0	0	0	
$x^{(1)}$	2.5	2.090 909	1.227 273	
$x^{(2)}$	2.977 273	2.028 926	1.004 132	
$x^{(3)}$	3.009 814	1.996 807	0.995 891	
$x^{(4)}$	2.999 830	1.999 688	1.000 163	
$x^{(5)}$	2.999 842	2.000 072	1.000 061	

从表 4－2 可以看出，用 Gauss－Seidel 迭代式(4－12)求解时，迭代向量 $x^{(k)}$ 逐步逼近 $x^*=(3,2,1)^T$ 的速度比 Jacobi 方法更快，且

$$\|x^{(5)}-x^*\|_\infty=1.58\times10^{-4}<0.5\times10^{-3}.$$

终止法则：因为

$$\|x^{(5)}-x^{(4)}\|_\infty=\max_{1\leq i\leq 3}|x_i^{(5)}-x_i^{(4)}|$$
$$=0.384\times10^{-3}<0.5\times10^{-3},$$

所以，$x^*\approx x^{(5)}=(2.999\,842,2.000\,072,1.000\,061)^T\approx(3,2,1)^T$.

我们要指出，表 4－2 与表 4－1 的表头虽然一样，但每次迭代计算的方法却不一样，在表 4－2 中我们采用了逐个替换分量的办法. 下面以第一次迭代为例说明如何利用 $x^{(0)}$ 计算 $x^{(1)}$.

$$x_1^{(1)}=(B_J)_{1\times}x^{(0)}+d_{J1}$$
$$=(0,3/8,-1/4)\begin{bmatrix}0\\0\\0\end{bmatrix}+2.5=2.5,$$

$$x_2^{(1)}=(B_J)_{2\times}\begin{bmatrix}x_1^{(1)}\\x_2^{(0)}\\x_3^{(0)}\end{bmatrix}+d_{J2}$$
$$=(-4/11,0,1/11)\begin{bmatrix}2.5\\0\\0\end{bmatrix}+3=2.090\,909,$$

$$x_3^{(1)}=(B_J)_{3*}\begin{bmatrix}x_1^{(1)}\\x_2^{(1)}\\x_3^{(0)}\end{bmatrix}+d_{J3}$$

$$=(-1/2,-1/4,0)\begin{bmatrix}2.5\\2.090\,909\\0\end{bmatrix}+3=1.227\,273.$$

4.2.3.5 算法设计

(1)数据存储方式.

$A[1:n,1:n]$,二维数组,存放系数矩阵的元素;

$b[1:n]$,一维数组,存放右端向量的分量;

$x[1:n]$,一维数组,既存放迭代当前向量 $x^{(k)}$,也存放迭代后继向量 $x^{(k+1)}$.

(2)迭代公式.

$$x_i \leftarrow (b_i-\sum_{\substack{j=1\\j\neq i}}^{n}a_{ij}x_j)/a_{ii},\quad i=1,2,\cdots,n,$$

$$\begin{bmatrix}x_1\\x_2\\\vdots\\x_n\end{bmatrix}\begin{matrix}\leftarrow & x_1 & x_2 & \cdots & x_n\\\leftarrow & x_1 & x_2 & \cdots & x_n\\ & \vdots & \vdots & & \vdots\\\leftarrow & x_1 & x_2 & \cdots & x_n\end{matrix}$$

每一次迭代完毕,仍以 x 作为下一次迭代的当前向量.

(3)终止法则.

Jacobi 迭代法:

检查:$\|x-y\|_\infty=\max\limits_{1\leq i\leq n}|x_i-y_i|<\varepsilon$?

是,$x\Rightarrow x^*$, stop;

否,$x_i\Rightarrow y_i(i=1,2,\cdots,n)$,继续执行下一次迭代.

Gauss-Seidel 迭代法:

令 $e=\|x^{(k+1)}-x^{(k)}\|_\infty$,

检查:$e<\varepsilon$?

是,$x\Rightarrow x^*$, stop;

否,$x_i^{(k+1)}\Rightarrow x_i^{(k)}(i=1,2,\cdots,n)$,继续执行下一次迭代.

(4)算法.

Algo. 4.2(设已确知算法收敛)

1° 读入 A,b,ε.

2° 送初始向量 $0\to x_i(i=1,2,\cdots,n)$.

3° 送偏差初值 $0\to e$.

4° $1\to i$(循环变量赋初值).

5° $x_i\to t$(t 为工作单元).

6° 计算迭代值$(b_i-\sum_{\substack{j=1\\j\neq i}}^{n}a_{ij}x_j)/a_{ii}\to x_i$.

7° 判断:$|x_i-t|\leqslant e$?

是,转 8°;

否,$|x_i-t|\to e$,转 8°.

8° 检查:$i=n$?

是,转 9°(当前迭代完毕);

否,$i+1\to i$,转 9°(继续完成当前迭代).

9° 检查:$e<\varepsilon$?

是,打印 $x_i(i=1,2,\cdots,n)$, stop;

否,转 3°出口处,继续执行下一次迭代.

注 算法中设置两个工作单元 t 及 e,其中 t 存放 $x_i^{(k)}$ 的值;e 开始存放偏差初值,迭代过程中存放 $\|\boldsymbol{x}^{(k+1)}-\boldsymbol{x}^{(k)}\|_\infty$,即

$$e=\max_{1\leqslant i\leqslant n}|x_i^{(k+1)}-x_i^{(k)}|.$$

4.2.3.6 Gauss-Seidel 迭代的同步形式

前面我们分别介绍了 Gauss-Seidel 迭代式的两种表达形式,即分量形式和矩阵形式,它们都是异步迭代式,但在以后讨论迭代式的收敛性时需要用到该迭代式的同步形式.

设
$$\begin{aligned}\boldsymbol{x}^{(k+1)}&=\boldsymbol{B}_{J1}\boldsymbol{x}^{(k+1)}+\boldsymbol{B}_{J2}\boldsymbol{x}^{(k)}+\boldsymbol{d}_J\\&=-\boldsymbol{D}^{-1}\boldsymbol{L}\boldsymbol{x}^{(k+1)}-\boldsymbol{D}^{-1}\boldsymbol{U}\boldsymbol{x}^{(k)}+\boldsymbol{D}^{-1}\boldsymbol{b}.\end{aligned}$$

作如下变形

$$(\boldsymbol{I}+\boldsymbol{D}^{-1}\boldsymbol{L})\boldsymbol{x}^{(k+1)}=-\boldsymbol{D}^{-1}\boldsymbol{U}\boldsymbol{x}^{(k)}+\boldsymbol{D}^{-1}\boldsymbol{b},$$
$$\boldsymbol{D}^{-1}(\boldsymbol{D}+\boldsymbol{L})\boldsymbol{x}^{(k+1)}=-\boldsymbol{D}^{-1}\boldsymbol{U}\boldsymbol{x}^{(k)}+\boldsymbol{D}^{-1}\boldsymbol{b},$$
$$\boldsymbol{x}^{(k+1)}=-(\boldsymbol{D}+\boldsymbol{L})^{-1}\boldsymbol{U}\boldsymbol{x}^{(k)}+(\boldsymbol{D}+\boldsymbol{L})^{-1}\boldsymbol{b}.$$

令
$$\boldsymbol{B}_S=-(\boldsymbol{D}+\boldsymbol{L})^{-1}\boldsymbol{U},$$
$$\boldsymbol{d}_S=(\boldsymbol{D}+\boldsymbol{L})^{-1}\boldsymbol{b}.$$

由此得 Gauss-Seidel 迭代公式的同步形式

$$\begin{cases}\boldsymbol{x}^{(k+1)}=\boldsymbol{B}_S\boldsymbol{x}^{(k)}+\boldsymbol{d}_S,\\\boldsymbol{B}_S=-(\boldsymbol{D}+\boldsymbol{L})^{-1}\boldsymbol{U},\\\boldsymbol{d}_S=(\boldsymbol{D}+\boldsymbol{L})^{-1}\boldsymbol{b}.\end{cases}$$

4.2.4 迭代公式归纳

4.2.4.1 Jacobi 迭代公式

(1)矩阵形式(用于表格计算,收敛性判定)

$$\begin{cases}\boldsymbol{x}^{(k+1)}=\boldsymbol{B}_J\boldsymbol{x}^{(k)}+\boldsymbol{d}_J,\\\boldsymbol{B}_J=-\boldsymbol{D}^{-1}(\boldsymbol{L}+\boldsymbol{U}),\\\boldsymbol{d}_J=\boldsymbol{D}^{-1}\boldsymbol{b}.\end{cases}$$

(2)分量形式(用于非表格计算)

$$x_i^{(k+1)}=(b_i-\sum_{\substack{j=1\\j\neq i}}^{n}a_{ij}x_j^{(k)})/a_{ii},\quad i=1,2,\cdots,n.$$

(3)存储形式(用于算法设计)

$$x_i=(b_i-\sum_{\substack{j=1\\j\neq i}}^{n}a_{ij}y_j)/a_{ii},\quad i=1,2,\cdots,n.$$

4.2.4.2 Gauss-Seidel 迭代公式

(1)矩阵异步形式(用于表格计算)

$$\begin{cases}\boldsymbol{x}^{(k+1)}=\boldsymbol{B}_{J1}\boldsymbol{x}^{(k+1)}+\boldsymbol{B}_{J2}\boldsymbol{x}^{(k)}+\boldsymbol{d}_J,\\ \boldsymbol{B}_{J1}=-\boldsymbol{D}^{-1}\boldsymbol{L},\quad \boldsymbol{B}_{J2}=-\boldsymbol{D}^{-1}\boldsymbol{U},\\ \boldsymbol{d}_J=\boldsymbol{D}^{-1}\boldsymbol{b}.\end{cases}$$

(2)矩阵同步形式(用于收敛性判定)

$$\begin{cases}\boldsymbol{x}^{(k+1)}=\boldsymbol{B}_S\boldsymbol{x}^{(k)}+\boldsymbol{d}_S,\\ \boldsymbol{B}_S=-(\boldsymbol{D}+\boldsymbol{L})^{-1}\boldsymbol{U},\\ \boldsymbol{d}_S=(\boldsymbol{D}+\boldsymbol{L})^{-1}\boldsymbol{b}.\end{cases}$$

(3)分量形式(用于非表格计算)

$$x_i^{(k+1)}=(b_i-\sum_{j=1}^{i-1}a_{ij}x_j^{(k+1)}-\sum_{j=i+1}^{n}a_{ij}x_j^{(k)})/a_{ii},\quad i=1,2,\cdots,n.$$

(4)存储形式(用于算法设计)

$$x_i\leftarrow(b_i-\sum_{\substack{j=1\\j\neq i}}^{n}a_{ij}x_j)/a_{ii},\quad i=1,2,\cdots,n.$$

4.3 迭代法的收敛性

4.3.1 迭代法收敛的充要条件

定理 4.4 设

$$\boldsymbol{x}^{(k+1)}=\boldsymbol{B}\boldsymbol{x}^{(k)}+\boldsymbol{d},\tag{4-13}$$

其中 $\boldsymbol{B}$ 为迭代矩阵($\boldsymbol{B}_J$,$\boldsymbol{B}_S$ 仅为它的特例).

对 $\forall \boldsymbol{x}^{(0)}\in R^n$,迭代公式(4-13)收敛的充分必要条件是 $\rho(\boldsymbol{B})<1$.

证 设对 $\forall \boldsymbol{x}^{(0)}$,由式(4-13)所产生的迭代向量序列为$\{\boldsymbol{x}^{(k)}\}$,记 $\boldsymbol{\varepsilon}^{(k)}$ 为 $\boldsymbol{x}^{(k)}$ 对于极限向量 $\boldsymbol{x}^*$ 的偏差向量,即

$$\boldsymbol{\varepsilon}^{(k)}=\boldsymbol{x}^{(k)}-\boldsymbol{x}^*.$$

因为 $$\boldsymbol{x}^{(k)}=\boldsymbol{B}\boldsymbol{x}^{(k-1)}+\boldsymbol{d},\quad \boldsymbol{x}^*=\boldsymbol{B}\boldsymbol{x}^*+\boldsymbol{d},$$

所以

$$\begin{aligned}\boldsymbol{\varepsilon}^{(k)}&=\boldsymbol{B}\boldsymbol{x}^{(k-1)}-\boldsymbol{B}\boldsymbol{x}^*\\&=\boldsymbol{B}(\boldsymbol{x}^{(k-1)}-\boldsymbol{x}^*)\\&=\boldsymbol{B}\boldsymbol{\varepsilon}^{(k-1)}.\end{aligned}\tag{4-14}$$

据式(4-14)反复递推得

$$\varepsilon^{(k)} = B^k \varepsilon^{(0)}.$$

故当 $k \to \infty$ 时，$\varepsilon^{(k)} \to \theta$ 的充分必要条件是 $B^{(k)} \to \theta$.

又据定理 4.3，当 $k \to \infty$ 时，$B^k \to \theta$ 的充分必要条件是 $\rho(B) < 1$. 所以，当 $k \to \infty$ 时，$\varepsilon^{(k)} \to 0$，即 $x^{(k)} \to x^*$ 的充分必要条件是 $\rho(B) < 1$.

4.3.2 迭代法收敛的充分条件(一)

定理 4.5 设 $x^{(k+1)} = Bx^{(k)} + d$，若 $\| B \| = q < 1$，则有

(1)对 $\forall x^{(0)}$，$\lim\limits_{k \to \infty} x^{(k)} = x^*$； (4－15)

(2) $\| x^{(k)} - x^* \| \leqslant \frac{1}{1-q} \| x^{(k+1)} - x^{(k)} \|$. (4－16)

证 (1)因为 $\| B \| = q < 1$，据定理 4.2，$\rho(B) \leqslant \| B \| = q < 1$，所以据定理 4.4，对 $\forall x^{(0)}$，式(4－15)收敛，即

$$\lim_{k \to \infty} x^{(k)} = x^*, \quad x^* = Bx^* + d.$$

(2)因为

$$x^{(k+1)} - x^{(k)} = (x^* - x^{(k)}) - (x^* - x^{(k+1)}),$$

所以

$$\begin{aligned} \| x^{(k+1)} - x^{(k)} \| &= \| (x^* - x^{(k)}) - (x^* - x^{(k+1)}) \| \\ &\geqslant \| x^* - x^{(k)} \| - \| x^* - x^{(k+1)} \|. \end{aligned} \tag{4-17}$$

而

$$x^* - x^{(k+1)} = B(x^* - x^{(k)}),$$

故

$$\begin{aligned} \| x^* - x^{(k+1)} \| &= \| B(x^* - x^{(k)}) \| \\ &\leqslant \| B \| \, \| x^* - x^{(k)} \| \\ &= q \| x^* - x^{(k)} \|. \end{aligned} \tag{4-18}$$

比较(4－17)与(4－18)两式得

$$\begin{aligned} \| x^{(k+1)} - x^{(k)} \| &\geqslant \| x^* - x^{(k)} \| - q \| x^* - x^{(k)} \| \\ &= (1-q) \| x^* - x^{(k)} \|. \end{aligned}$$

故有

$$\| x^* - x^{(k)} \| \leqslant \frac{1}{1-q} \| x^{(k+1)} - x^{(k)} \|,$$

即式(4－16)成立. 不等式(4－16)与第 2 章中的不等式(2－9)不仅形式极为相似，其作用也是一致的.

由 $q < 1$，$0 < 1-q < 1$，当

$$\| x^{(k+1)} - x^{(k)} \| < (1-q)\varepsilon = \varepsilon',$$

有

$$\| x^* - x^{(k)} \| < \varepsilon.$$

这就是迭代法终止法则的理论依据，由此形成迭代误差的事后估计方法.

例 6 就 A_1，A_2 为系数矩阵构成的线性方程组，讨论 Jacobi 迭代法与 Gauss－Seidel 迭代法的收敛性. 其中

$$(1) A_1 = \begin{bmatrix} 10 & -2 & -1 \\ -2 & 10 & 1 \\ -1 & -2 & 5 \end{bmatrix}; \qquad (2) A_2 = \begin{bmatrix} 4 & -2 & -1 \\ -2 & 4 & -2 \\ -1 & -2 & 3 \end{bmatrix}.$$

解 (1)由 $A_1 = \begin{bmatrix} 10 & -2 & -1 \\ 2 & 10 & -1 \\ 1 & 2 & 5 \end{bmatrix}$ 计算 Jacobi 迭代矩阵

$$B_J=-D^{-1}(L+U)=\begin{bmatrix}0 & 0.2 & 0.1\\0.2 & 0 & 0.1\\0.2 & 0.4 & 0\end{bmatrix}.$$

因为 $\|B_J\|_\infty=\max(0.3,0.3,0.6)=0.6<1$,

所以,据定理 4.5,Jacobi 迭代式收敛.

由 A_1 计算 Gauss-Seidel 迭代矩阵

$$B_S=-(D+L)^{-1}U=\begin{bmatrix}0 & 0.2 & 0.1\\0 & 0.04 & 0.12\\0 & 0.056 & 0.068\end{bmatrix}.$$

因为 $\|B_S\|_\infty=0.3<1$,

所以,据定理 4.5,Gauss-Seidel 迭代式收敛.

(2)由 $A_2=\begin{bmatrix}4 & -2 & -1\\-2 & 4 & -2\\-1 & -2 & 3\end{bmatrix}$,计算 Jacobi 迭代矩阵

$$B_J=-D^{-1}(L+U)=\begin{bmatrix}0 & 1/2 & 1/4\\1/2 & 0 & 1/2\\1/3 & 2/3 & 0\end{bmatrix}.$$

因为 $\|B_J\|_\infty=1,\quad \|B_J\|_1=7/6$,

故定理 4.5 判别失效,改用定理 4.4 来判别:

$$\det(\lambda I-B_J)=\begin{vmatrix}\lambda & -1/2 & -1/4\\-1/2 & \lambda & -1/2\\-1/3 & -2/3 & \lambda\end{vmatrix}$$

$$=\lambda^3-\frac{2}{3}\lambda-\frac{1}{6}=0,$$

即 $6\lambda^3-4\lambda-1=0.$

用 Newton 迭代法求出该方程的三个根为

$$\lambda_1=0.920\,7,\quad \lambda_2=-0.284\,6,\quad \lambda_3=-0.636\,1.$$

故 $\rho(B_J)=0.920\,7<1$,

据定理 4.4,Jacobi 迭代式收敛.

又由 $$B_S=-(D+L)^{-1}U=\begin{bmatrix}0 & 1/2 & 1/4\\0 & 1/4 & 5/8\\0 & 1/3 & 1/2\end{bmatrix},$$

$$\|B_S\|_\infty=0.875<1,$$

据定理 4.5,Gauss-Seidel 迭代式收敛.

在本例中,判别 Gauss-Seidel 迭代式的收敛性要涉及迭代矩阵 $B_S=-(D+L)^{-1}U$. 若用定理 4.5 判别,要先计算 B_S,再计算 $\|B_S\|$,而计算逆矩阵 $(D+L)^{-1}$ 是很不方便的;若用定理 4.4 判别,同样要先计算 B_S,再计算 $\rho(B_S)$. 为了避免计算逆矩阵,我们证明在一定条件下,以下两个非线性方程是等价的.

$$\det(\lambda I - B_S) = \det[\lambda I + (D+L)^{-1}U] = 0. \tag{4-19}$$

$$\det[(\lambda(D+L)+U] = 0. \tag{4-20}$$

事实上，
$$\begin{aligned}\det(\lambda I - B_S) &= \det[\lambda I + (D+L)^{-1}U] \\ &= \det(D+L)^{-1} \cdot \det[\lambda(D+L)+U].\end{aligned}$$

当 $\det(D+L)^{-1} \neq 0$ 时，有

$$\det(\lambda I - B_S) = 0 \Leftrightarrow \det[\lambda(D+L)+U] = 0.$$

我们知道，求解式(4－20)较之式(4－19)容易得多.

首先以例6中 A_1 为例，对应的Gauss－Seidel迭代矩阵为 $B_S = -(D+L)^{-1}U$. 下面不用式(4－19)，而用式(4－20)来求解 B_S 的全部特征值.

因为
$$\det[\lambda(D+L)+U] = \begin{vmatrix} 10\lambda & -2 & -1 \\ -2\lambda & 10\lambda & -1 \\ -\lambda & -2\lambda & 5\lambda \end{vmatrix} = \lambda(500\lambda^2 - 54\lambda - 2) = 0,$$

解得
$$\lambda_1 = 0, \quad \lambda_{2,3} = \frac{27 \pm \sqrt{1\,729}}{500},$$

所以
$$\rho(B_S) = \frac{27 + \sqrt{1\,729}}{500} = 0.137\,2 < 1.$$

据定理4.4，Gauss－Seidel迭代式收敛.

再以例6中 A_2 为例，对应的Gauss－Seidel迭代矩阵为 $B_S = -(D+L)^{-1}U$，仍用式(4－20)求解 B_S 的全部特征值.

因为
$$\det[\lambda(D+L)+U] = \begin{vmatrix} 4\lambda & -2 & -1 \\ -2\lambda & 4\lambda & -2 \\ -\lambda & -2\lambda & 3\lambda \end{vmatrix} = 4\lambda(12\lambda^2 - 9\lambda - 1) = 0,$$

解得
$$\lambda_1 = 0, \quad \lambda_{2,3} = \frac{9 \pm \sqrt{129}}{24},$$

所以
$$\rho(B_S) = \frac{9 - \sqrt{129}}{24} = 0.848\,2 < 1.$$

据定理4.4，Gauss－Seidel迭代式收敛.

4.3.3 迭代法收敛的充分条件(二)

定义4.7(强超条件与弱超条件)　若 $A = (a_{ij})_{n \times n}$ 满足

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|, \quad i = 1, 2, \cdots, n, \tag{4-21}$$

则称 A 具有严格对角优势，或者说 A 满足主对角元强超条件(简称强超条件)；

若 A 满足
$$|a_{ii}| \geq \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|, \quad i = 1, 2, \cdots, n, \tag{4-22}$$

且至少有一个 i 值满足严格不等式，则称 A 具有对角优势，或者说 A 满足主对角元弱超条

件(简称弱超条件).

例 7 $A=\begin{bmatrix}8 & -3 & 2\\ 4 & 11 & -1\\ 6 & 3 & 12\end{bmatrix}$满足式(4-21),称 A 满足强超条件;

$B=\begin{bmatrix}4 & -2 & -1\\ -2 & 4 & -2\\ -1 & -2 & 3\end{bmatrix}$满足式(4-22),称 B 满足弱超条件;

同理,$C=\begin{bmatrix}-2 & 1 & 0 & 0\\ 1 & -2 & 1 & 0\\ 0 & 1 & -2 & 1\\ 0 & 0 & 1 & -2\end{bmatrix}$也满足弱超条件.

定义 4.8(不可约矩阵) 若 $A=(a_{ij})_{n\times n}$,通过行对换和相应的列对换变成

$$\begin{pmatrix}A_{11} & A_{12}\\ 0 & A_{22}\end{pmatrix},$$

其中 A_{11},A_{22}为方阵,则称 A 为可约矩阵;反之为不可约矩阵.例如:

$$\begin{bmatrix}a_{11} & \vdots & a_{12} & a_{13}\\ \cdots & \cdots & \cdots & \cdots\\ 0 & \vdots & a_{22} & a_{23}\\ 0 & \vdots & a_{32} & a_{33}\end{bmatrix}$$

为可约矩阵,

$$\begin{bmatrix}a_{11} & \vdots & a_{12} & a_{13} & a_{14}\\ a_{21} & \vdots & a_{22} & a_{23} & a_{24}\\ 0 & \vdots & a_{32} & a_{33} & a_{34}\\ 0 & \vdots & a_{42} & a_{43} & a_{44}\end{bmatrix}$$

为不可约矩阵.又如,在例 7 中,A 为强超不可约矩阵;B 为弱超不可约矩阵;C 为弱超不可约矩阵.

定理 4.6(非奇异条件) 设矩阵 $A=(a_{ij})_{n\times n}$.

(1)若 A 满足主对角元强超条件,则 $\det(A)\neq 0$;

(2)若 A 满足主对角元弱超条件且不可约,则 $\det(A)\neq 0$.

证 就情形(1)证明如下:

采用反证法,即设 $\det(A)=0$,则存在 $u\neq 0$ 使得 $Au=0$,记 $u=(u_1,u_2,\cdots,u_n)^T$,又记

$$|u_t|=\max_{1\leq i\leq n}|u_i|\neq 0. \tag{4-23}$$

由 $Au=0$ 的第 t 个方程有

$$\sum_{j=1}^{n} a_{tj}u_j=0,$$

即

$$a_{tt}u_t+\sum_{\substack{j=1\\ j\neq t}}^{n} a_{tj}u_j=0.$$

据式(4－23)有

$$|a_{ii}u_i| = |\sum_{\substack{j=1\\j\neq i}}^{n} a_{ij}u_j| \leqslant \sum_{\substack{j=1\\j\neq i}}^{n} |a_{ij}||u_j| \leqslant |u_i| \sum_{\substack{j=1\\j\neq i}}^{n} |a_{ij}|,$$

所以

$$|a_{ii}| \leqslant \sum_{\substack{j=1\\j\neq i}}^{n} |a_{ij}|. \tag{4-24}$$

式(4－24)与命题假设强超条件矛盾，故有 $\det(A)\neq 0$.

定理 4.7(收敛的充分条件)　设 $Ax=b$.

(1)若 A 满足主对角元强超条件，则对 $\forall x^{(0)}\in R^n$，Jacobi 迭代法及 Gauss－Seidel 迭代法均收敛；

(2)若 A 满足主对角元弱超条件且不可约，则对 $\forall x^{(0)}\in R^n$，Jacobi 迭代法及 Gauss－Seidel 迭代法均收敛.

证　仅就情形(1)证明如下：

首先证明 Jacobi 迭代式 $x^{(k+1)}=B_J x^{(k)}+d_J$ 收敛.

因为

$$B_J=\begin{bmatrix} 0 & -a_{12}/a_{11} & \cdots & a_{1n}/a_{11} \\ -a_{21}/a_{22} & 0 & \cdots & -a_{2n}/a_{22} \\ \vdots & \vdots & \ddots & \vdots \\ -a_{n1}/a_{nn} & \cdots & -a_{n,n-1}/a_{nn} & 0 \end{bmatrix},$$

$$\|B_J\|_\infty = \max_{1\leqslant i\leqslant n}\sum_{\substack{j=1\\j\neq i}}^{n} |a_{ij}| / |a_{ii}| = \max_{1\leqslant i\leqslant n}\frac{\sum_{\substack{j=1\\j\neq i}}^{n}|a_{ij}|}{|a_{ii}|},$$

由题设条件

$$|a_{ii}| > \sum_{\substack{j=1\\j\neq i}}^{n} |a_{ij}|,\quad i=1,2,\cdots,n,$$

亦即

$$|a_{ii}| > \max_{1\leqslant i\leqslant n}\frac{\sum_{\substack{j=1\\j\neq i}}^{n}|a_{ij}|}{|a_{ii}|}<1,$$

所以

$$\|B_J\|_\infty<1.$$

据定理 4.5，Jacobi 迭代式收敛.

再证明 Gauss－Seidel 迭代式 $x^{(k+1)}=B_S x^{(k)}+d_S$ 收敛.

如前所述，方程(4－19)，即

$$\det(\lambda I-B_S)=0$$

与方程(4－20)，即

$$\det[\lambda(D+L)+U]=0$$

是等价的. 只要证明满足方程(4－20)的 λ 必须满足 $|\lambda|<1$，即 $\rho(B_S)<1$ 即可.

仍用反证法，设满足方程(4－20)的 λ 其绝对值不小于 1，即 $|\lambda|\geqslant 1$，由题设条件

$$|a_{ii}| > \sum_{\substack{j=1\\j\neq i}}^{n} |a_{ij}|,\quad |\lambda|\geqslant 1,$$

有
$$|\lambda|\cdot|a_{ii}|>\sum_{\substack{j=1\\j\neq i}}^{n}|\lambda||a_{ij}|,$$

即
$$|\lambda a_{ii}|>\sum_{j=1}^{i-1}|\lambda a_{ij}|+\sum_{j=i+1}^{n}|a_{ij}|. \tag{4-25}$$

式(4－25)表明矩阵 $\lambda(\boldsymbol{D}+\boldsymbol{L})+\boldsymbol{U}$ 是严格对角优势阵(见注).

据定理 4.6,$\det[\lambda(\boldsymbol{D}+\boldsymbol{L}+\boldsymbol{U}]\neq0$,此结论与假设矛盾,由此得到满足方程(4－20)的 λ,其绝对值应小于 1,即 $|\lambda|<1$,亦即 $\rho(\boldsymbol{B}_S)<1$.

注

$$\lambda(\boldsymbol{D}+\boldsymbol{L})+\boldsymbol{U}=\begin{bmatrix}\lambda a_{11} & a_{12} & \cdots & \cdots & \cdots & a_{1n}\\ \lambda a_{12} & \lambda a_{22} & \cdots & \cdots & \cdots & a_{2n}\\ \vdots & \vdots & \ddots & & & \vdots\\ \lambda a_{i1} & \lambda a_{i2} & \cdots & \lambda a_{ii} & \cdots & a_{in}\\ \vdots & \vdots & & & \ddots & \vdots\\ \lambda a_{n1} & \lambda a_{n2} & \cdots & \cdots & \cdots & \lambda a_{nn}\end{bmatrix},$$

第 i 行$(i=1,2,\cdots,n)$为

$$(\lambda a_{i1},\cdots,\lambda a_{i,i-1},\lambda a_{ii},a_{i,i+1},\cdots,a_{in}).$$

由式(4－25)
$$|\lambda a_{ii}|>\sum_{j=1}^{i-1}|\lambda a_{ij}|+\sum_{j=i+1}^{n}|a_{ij}|,$$

知 $\lambda(\boldsymbol{D}+\boldsymbol{L})+\boldsymbol{U}$ 是严格对角优势阵.

例 8 以下列矩阵作为线性方程组的系数矩阵,试判别它们对 Jacobi 迭代法,Gauss－Seidel 迭代法的收敛性.

(1)$\boldsymbol{A}_1=\begin{bmatrix}5&2&1\\1&3&2\\1&1&2\end{bmatrix}$.

解 该矩阵弱超不可约,故据定理 4.7,以它为系数矩阵构成线性方程组,对 $\forall\boldsymbol{x}^{(0)}$,Jacobi 方法与 Gauss－Seidel 迭代方法均收敛.

(2)$\boldsymbol{A}_2=\begin{bmatrix}1&-2&2\\-1&1&1\\-2&-2&1\end{bmatrix}$.

解 该矩阵不满足强超或弱超条件,不能应用定理 4.7,可考虑用定理 4.4 或定理 4.5.

对 Jacobi 迭代法,设 $\boldsymbol{A}_2$ 的迭代矩阵为 $\boldsymbol{B}_J$,且

$$\boldsymbol{B}_J=\begin{bmatrix}0&2&-2\\1&0&-1\\2&2&0\end{bmatrix},$$

$$\|\boldsymbol{B}_J\|_\infty>1,\quad \|\boldsymbol{B}\|_1>1,$$

$$\det(\lambda\boldsymbol{I}-\boldsymbol{B}_J)=\begin{vmatrix}\lambda&-2&2\\-1&\lambda&1\\-2&-2&\lambda\end{vmatrix}=\lambda^3+4\lambda+8.$$

令 $$f(\lambda)=\lambda^3+4\lambda+8.$$

因为 $f(-2)<0, f(-1)>0$,所以 $\exists\lambda\in(-2,-1)$,故 $\rho(B_J)>1$.

据定理 4.4,对 $\forall x^{(0)}$,Jacobi 迭代法不收敛.

对 Gauss-Seidel 迭代法,设 A_2 的迭代矩阵为 B_S,即

$$B_S=-(D+L)^{-1}U.$$

据式(4-20) $$\det[\lambda(D+L)+U]=0,$$

即 $$\begin{vmatrix}\lambda & -2 & 2\\ -\lambda & \lambda & 1\\ -2\lambda & -2\lambda & \lambda\end{vmatrix}=\lambda^3+8\lambda^2+4\lambda=\lambda(\lambda^2+8\lambda+4)=0,$$

求解得 $$\lambda_1=0,\quad \lambda_{2,3}=\frac{-8\pm\sqrt{48}}{2}.$$

所以,$\rho(B_S)>1$.据定理 4.4,Gauss-Seidel 迭代法对 $\forall x^{(0)}$ 均不收敛.

(3) $A_3=\begin{bmatrix}-2 & 1 & 0 & 0\\ 1 & -2 & 1 & 0\\ 0 & 1 & -2 & 1\\ 0 & 0 & 1 & -2\end{bmatrix}$.

解 该矩阵弱超不可约,对两种迭代法均收敛.

(4) $A_4=\begin{bmatrix}5 & -1 & -1 & -1\\ -1 & 10 & -1 & -1\\ -1 & -1 & 5 & -1\\ -1 & -1 & -1 & 10\end{bmatrix}$.

解 该矩阵强超,对两种迭代法均收敛.

(5) $A_5=\begin{bmatrix}1 & 1/2 & 1/2\\ 1/2 & 1 & 1/2\\ 1/2 & 1/2 & 1\end{bmatrix}$.

解 该矩阵不满足强超或弱超条件.

首先考虑 Jacobi 迭代法,设其迭代矩阵为 B_J,且

$$B_J=\begin{bmatrix}0 & -1/2 & -1/2\\ -1/2 & 0 & -1/2\\ -1/2 & -1/2 & 0\end{bmatrix},$$

$$\|B_J\|_\infty=\|B_J\|_1=1,$$

$$\det(\lambda I-B_J)=\begin{vmatrix}\lambda & 1/2 & 1/2\\ 1/2 & \lambda & 1/2\\ 1/2 & 1/2 & \lambda\end{vmatrix}=\lambda^3-\frac{3}{4}\lambda+\frac{1}{4}=0,$$

即 $$4\lambda^3-3\lambda+1=(\lambda+1)(4\lambda^2-4\lambda+1)=0,$$

求解得 $$\lambda_1=-1,\quad \rho(B_J)=1.$$

据定理 4.4 知,Jacobi 迭代法不收敛.

其次考虑 Gauss-Seidel 迭代法,设其迭代矩阵为 B_S,则

$$B_S=-(D+L)^{-1}U.$$

据方程(4－20)

$$\det[\lambda(D+L)+U]=0,$$

即
$$\begin{vmatrix} \lambda & 1/2 & 1/2 \\ (1/2)\lambda & \lambda & 1/2 \\ (1/2)\lambda & (1/2)\lambda & \lambda \end{vmatrix}=\lambda^3-\frac{5}{8}\lambda^2+\frac{1}{8}\lambda=\lambda(\lambda^2-\frac{5}{8}\lambda+\frac{1}{8})=0,$$

解得
$$\lambda_1=0,\quad \lambda_{2,3}=\frac{5\pm\sqrt{-7}}{16}=\frac{5}{16}\pm\frac{\sqrt{7}}{16}i,$$

即
$$\rho(B_S)<1.$$

据定理 4.4 知，对 $\forall x^{(0)}$ Gauss－Seidel 迭代法收敛.

4.3.4 迭代法收敛性判别归纳

设 $Ax=b$，相应的迭代公式为 $x^{(k+1)}=Bx^{(k)}+d$，其中 $B=B_J$ 或 B_S.

(1)检查：A 是否满足强超条件(或弱超不可约条件)？

　　是，$x^{(k+1)}=Bx^{(k)}+d$，对 $\forall x^{(0)}$ 均收敛，　stop；

　　否，转(2).

(2)$Ax=b\xrightarrow{\text{换行}}A'x=b'$，检查：$A'$是否满足强超条件或弱超不可约条件？

　　是，$x^{(k+1)}=B'x^{(k)}+d'$(B'是A'对应的迭代矩阵)，对 $\forall x^{(0)}$ 均收敛，　stop；

　　否，由(1)直接转(3).

(3)检查：$\|B\|<1$？

　　是，$x^{(k+1)}=Bx^{(k)}+d$，对 $\forall x^{(0)}$ 均收敛，　stop；

　　否，转(4).

(4)检查：$\rho(B)<1$？

　　是，$x^{(k+1)}=Bx^{(k)}+d$，对 $\forall x^{(0)}$ 均收敛，　stop；

　　否，该迭代式不收敛.

4.4 逐次超松驰方法(SOR 方法)

4.4.1 SOR 公式

SOR 公式又称为加速的 Gauss－Seidel 公式，分两步构成：

(1)迭代步，即用 Gauss－Seidel 公式给出预报值$\bar{x}_i^{(k+1)}$.

$$\bar{x}_i^{(k+1)}=(b_i-\sum_{j=1}^{i-1}a_{ij}x_j^{(k+1)}-\sum_{j=i+1}^{n}a_{ij}x_j^{(k)})/a_{ii},\quad i=1,2,\cdots,n.$$

(2)加速步，即将当前迭代值 $x_i^{(k)}$ 与预报值$\bar{x}_i^{(k+1)}$ 线性组合得校正值 $x_i^{(k+1)}$.

$$x_i^{(k+1)}=\omega\bar{x}_i^{(k+1)}+(1-\omega)x_i^{(k)},\quad i=1,2,\cdots,n.$$

其中 $\omega\in(0,2)$称为松驰因子.

　　合并以上两式得

$$
\begin{aligned}
x_i^{(k+1)} &= x_i^{(k)} + \omega(\overline{x}_i^{(k+1)} - x_i^{(k)}) \\
&= x_i^{(k)} + \omega\left[\overline{x}_i^{(k+1)} - \frac{a_{ii}}{a_{ii}} x_i^{(k)}\right] \\
&= x_i^{(k)} + \frac{\omega}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i}^{n} a_{ij} x_j^{(k)}\right), \quad i = 1,2,\cdots,n.
\end{aligned} \tag{4-26}
$$

或者写成

$$
\begin{cases}
x_i^{(k+1)} = x_i^{(k)} + \Delta x_i^{(k)}, \\
\Delta x_i^{(k)} = \dfrac{\omega}{a_{ii}}\left(b_i - \displaystyle\sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i}^{n} a_{ij} x_j^{(k)}\right), \quad i = 1,2,\cdots n.
\end{cases} \tag{4-27}
$$

我们称式(4-26)或者式(4-27)为 SOR 公式；当 $\omega \in (0,1)$ 时，称为松弛公式；当 $\omega \in (1,2)$ 时，称为超松驰公式；特别地，当 $\omega = 1$ 时，则为 Gauss-Seidel 公式.

4.4.2 SOR 公式的矩阵形式

将式(4-27)中的增量 $\Delta x_i^{(k)}$ 改写成

$$
\Delta x_i^{(k)} = \omega\left[-\sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j^{(k+1)} - \sum_{j=i+1}^{n} \frac{a_{ij}}{a_{ii}} x_j^{(k)} - \frac{a_{ii}}{a_{ii}} x_i^{(k)} + \frac{b_i}{a_{ii}}\right], \quad i = 1,2,\cdots,n.
$$

由此得向量 $\Delta \boldsymbol{x}^{(k)}$ 的表达式

$$
\Delta \boldsymbol{x}^{(k)} = \omega(\boldsymbol{B}_{J1} \boldsymbol{x}^{(k+1)} + \boldsymbol{B}_{J2} \boldsymbol{x}^{(k)} - \boldsymbol{I} \boldsymbol{x}^{(k)} + \boldsymbol{d}_J).
$$

据式(4-27)可得

$$
\begin{cases}
\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \Delta \boldsymbol{x}^{(k)}, \\
\Delta \boldsymbol{x}^{(k)} = \omega(\boldsymbol{B}_{J1} \boldsymbol{x}^{(k+1)} + \boldsymbol{B}_{J2}{}^{(k)} - \boldsymbol{I} \boldsymbol{x}^{(k)} + \boldsymbol{d}_J), \\
\boldsymbol{B}_{J1} = -\boldsymbol{D}^{-1}\boldsymbol{L}, \quad \boldsymbol{B}_{J2} = -\boldsymbol{D}^{-1}\boldsymbol{U}, \quad \boldsymbol{d}_J = \boldsymbol{D}^{-1}\boldsymbol{b}.
\end{cases} \tag{4-28}
$$

称式(4-28)为 SOR 公式的异步矩阵形式.

为便于判别 SOR 公式的收敛性，我们还可将式(4-28)改写成同步形式

$$
\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \omega \boldsymbol{B}_{J1} \boldsymbol{x}^{(k+1)} + \omega \boldsymbol{B}_{J2} \boldsymbol{x}^{(k)} - \omega \boldsymbol{x}^{(k)} + \omega \boldsymbol{d}_J,
$$

$$
(\boldsymbol{I} - \omega \boldsymbol{B}_{J1}) \boldsymbol{x}^{(k+1)} = (1-\omega) \boldsymbol{x}^{(k)} + \omega \boldsymbol{B}_{J2} \boldsymbol{x}^{(k)} + \omega \boldsymbol{d}_J.
$$

将

$$
\boldsymbol{B}_{J1} = -\boldsymbol{D}^{-1}\boldsymbol{L}, \quad \boldsymbol{B}_{J2} = -\boldsymbol{D}^{-1}\boldsymbol{U}, \quad \boldsymbol{d}_J = \boldsymbol{D}^{-1}\boldsymbol{b}
$$

代入上式并化简得

$$
\begin{cases}
\boldsymbol{x}^{(k+1)} = \boldsymbol{G}_\omega \boldsymbol{x}^{(k)} + \boldsymbol{f}, \\
\boldsymbol{G}_\omega = (\boldsymbol{D} + \omega \boldsymbol{L})^{-1}[(1-\omega)\boldsymbol{D} - \omega \boldsymbol{U}], \\
\boldsymbol{f} = \omega(\boldsymbol{D} + \omega \boldsymbol{L})^{-1} \boldsymbol{b}.
\end{cases} \tag{4-29}
$$

称式(4-29)为 SOR 公式的同步矩阵形式.

4.4.3 SOR 方法的计算表格

根据式(4－28)作迭代计算，其计算格式见表 4－3.

表 4－3

$x^{(k)}$	$\omega(B_J-I)$				ωd_J
	$\omega(B_J-I)_{1\leftarrow}$				ωd_{J1}
	$\omega(B_J-I)_{2\leftarrow}$				ωd_{J2}
	⋮				⋮
	$\omega(B_J-I)_{n\leftarrow}$				ωd_{Jn}
$x^{(0)}$	$x_1^{(0)}$	$x_2^{(0)}$	…	$x_n^{(0)}$	
$\Delta x^{(0)}$	$\Delta x_1^{(0)}$	$\Delta x_2^{(0)}$	…	$\Delta x_n^{(0)}$	
$x^{(1)}=x^{(0)}+\Delta x^{(0)}$	$x_1^{(1)}$	$x_2^{(1)}$	…	$x_n^{(1)}$	
$\Delta x^{(1)}$	$\Delta x_1^{(1)}$	$\Delta x_2^{(1)}$	…	$\Delta x_n^{(1)}$	
$x^{(2)}=x^{(1)}+\Delta x^{(1)}$	$x_1^{(2)}$	$x_2^{(2)}$	…	$x_n^{(2)}$	
⋮	⋮	⋮		⋮	

与 Gauss－Seidel 迭代法的计算表格 4－2 一样，在上述表格的第二栏中我们列出矩阵$\omega(B_J-I)$，而没有列出 $\omega(B_{J1}+B_{J2}-I)$，但在每一步迭代中要按照分量逐个替换的方法进行计算. 下面以第一次迭代为例加以说明：

$$\begin{cases}\Delta x_1^{(0)}=\omega(B_J-I)_{1\leftarrow}\begin{bmatrix}x_1^{(0)}\\x_2^{(0)}\\\vdots\\x_n^{(0)}\end{bmatrix}+\omega d_{J1},\\ x_1^{(1)}=x_1^{(0)}+\Delta x_1^{(0)};\end{cases}$$

$$\begin{cases}\Delta x_2^{(0)}=\omega(B_J-I)_{2\leftarrow}\begin{bmatrix}x_1^{(1)}\\x_2^{(0)}\\\vdots\\x_n^{(0)}\end{bmatrix}+\omega d_{J2},\\ x_2^{(1)}=x_2^{(0)}+\Delta x_2^{(0)};\end{cases}$$

$$\vdots$$

$$\begin{cases}\Delta x_n^{(0)}=\omega(B_J-I)_{n\leftarrow}\begin{bmatrix}x_1^{(1)}\\x_2^{(1)}\\\vdots\\x_{n-1}^{(1)}\\x_n^{(0)}\end{bmatrix}+\omega d_{Jn},\\ x_n^{(1)}=x_n^{(0)}+\Delta x_n^{(0)}.\end{cases}$$

4.4.4 SOR 方法的收敛性

讨论 SOR 方法的收敛性，宜采用同步公式(4−29)$x^{(k+1)}=G_\omega x^{(k)}+f$. 鉴于该迭代公式是一般的迭代公式(4−13)$x^{(k+1)}=Bx^{(k)}+d$ 的特例，因此，判别收敛性的充要条件定理 4.4 及充分条件定理 4.5 都适用于 SOR 迭代式(4−29). 但是，迭代矩阵 G_ω 太复杂，使用定理 4.4 及定理 4.5 都不太方便. 下面我们针对迭代式(4−29)介绍一些实用的判别收敛性的方法.

定理 4.8(必要条件) 设 $Ax=b(a_{ii}\neq 0, i=1,2,\cdots,n)$，式(4−29)对 $\forall x^{(0)}$ 收敛的必要条件是

$$\omega\in(0,2).$$

证 设式(4−29)对 $\forall x^{(0)}$ 收敛，据定理 4.4，即有 $\rho(G_\omega)<1$. 又设 G_ω 的特征值为 λ_i $(i=1,2,\cdots,n)$，则相应的特征方程 $\det(G_\omega-\lambda I)=0$ 是含 λ 的 n 次代数方程，即

$$\lambda^n-p_1\lambda^{n-1}+p_2\lambda^{n-2}+\cdots+(-1)^n p_n=0.$$

据根与系数的关系(见第 5 章 5.1 预备知识)

$$\sum_{i=1}^n\lambda_i=p_1=\mathrm{tr}(G_\omega),$$

$$\prod_{i=1}^n\lambda_i=(-1)^n p_n=\det(G_\omega),$$

所以
$$|\det(G)|=\left|\prod_{i=1}^n\lambda_i\right|\leqslant[\rho(G_\omega)]^n,$$

即
$$|\det(G_\omega)|^{1/n}\leqslant\rho(G_\omega).$$

又据式(4−29)，

$$\det(G_\omega)=\det(D+\omega L)^{-1}\cdot\det[(1-\omega)D-\omega U]=(1-\omega)^n,$$

所以
$$|1-\omega|=|\det(G_\omega)|^{1/n}\leqslant\rho(G_\omega)<1,$$

即
$$\omega\in(0,2).$$

定理 4.8 的逆命题不一定成立，但逆否命题成立，即若松驰因子 $\omega\overline{\in}(0,2)$，则 SOR 迭代式(4−29)不收敛.

定理 4.9(充分条件) 设 $Ax=b, A^T=A, A>0$ 且 $\omega\in(0,2)$，则 SOR 迭代式(4−29)收敛(证明略).

推论 设 $Ax=b, A^T=A, A>0$，则 Gauss−Seidel 迭代法收敛.

实事上，只要在定理 4.9 中令 $\omega=1$，即得此推论.

例 9 用 SOR 方法解方程组 $Ax=b$，其中

$$A=\begin{bmatrix}-4&1&1&1\\1&-4&1&1\\1&1&-4&1\\1&1&1&-4\end{bmatrix},\quad b=(1,1,1,1)^T.$$

它的精确解为 $x^*=(-1,-1,-1,-1)^T$.

解 设初值 $x^{(0)}=\mathbf{0}$，表 4－4 列出了松弛因子 ω 取(0,2)内不同值时的计算结果.

表 4－4

松弛因子 ω	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9
迭代次数 $\\|x^{(k)}-x^*\\|_\infty<10^{-5}$	22	17	12	11	14	17	23	33	53	109

取 $\omega=1.3$，迭代 11 次的结果为

$$x^{(11)}=(-0.9999946,\ 1.00000310,\ -0.99999953,\ -0.99999912)^{\mathrm{T}},$$

$$\|x^{(11)}-x^*\|_\infty=0.354\times10^{-5}.$$

从表 4－4 可以看出，松弛因子选得好，会使 SOR 迭代法的收敛加速，本例中 $\omega=1.3$ 是最佳松弛因子.

*4.5 非线性方程组迭代解法简介

n 个变量 n 个方程的非线性方程组的一般形式为

$$\begin{cases} f_1(x_1,x_2,\cdots,x_n)=0 \\ f_2(x_1,x_2,\cdots,x_n)=0 \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ f_n(x_1,x_2,\cdots,x_n)=0 \end{cases}. \tag{4-30}$$

其中 $f_i(x_1,x_2,\cdots,x_n)(i=1,2,\cdots,n)$ 是 $x_1,x_2,\cdots,x_n$ 的非线性函数，若用向量记号，令

$$\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}},$$

$$\boldsymbol{F}(\boldsymbol{x})=(f_1(\boldsymbol{x}),f_2(\boldsymbol{x}),\cdots,f_n(\boldsymbol{x}))^{\mathrm{T}},$$

则式(4－30)变为

$$\boldsymbol{F}(\boldsymbol{x})=\mathbf{0}. \tag{4-31}$$

这里 F 表示定义在 R^n 中开域 D 上的非线性向量值函数，记为 $F:D\subset R^n\to R^n$.

若存在 $x^*=(x_1^*,x_2^*,\cdots,x_n^*)^{\mathrm{T}}$ 使得 $\boldsymbol{F}(x^*)=\mathbf{0}$，则称 x^* 为方程(4－31)的解.

4.5.1 直接迭代法(简单迭代法)

例 10 (引例) 设二元非线性方程组

$$\begin{cases} 0.1x_1^2-x_1+0.1x_2+1=0 \\ 0.1x_1-x_2+0.1x_2^2+2=0 \end{cases}. \tag{4-32}$$

令 $\boldsymbol{x}=(x_1,x_2)^{\mathrm{T}},\boldsymbol{F}(\boldsymbol{x})=(f_1(\boldsymbol{x}),f_2(\boldsymbol{x}))^{\mathrm{T}}$. 其中

$$f_1(\boldsymbol{x})=0.1x_1^2-x_1+0.1x_2+1,$$

$$f_2(\boldsymbol{x})=0.1x_1-x_2+0.1x_2^2+2,$$

则式(4－32)变为 $\boldsymbol{F}(\boldsymbol{x})=\mathbf{0}$.

现从式(4－32)的第一个方程中，虚拟解出 x_1：

$$x_1 = 0.1x_1^2 + 0.1x_2 + 1 \xlongequal{\text{令}} \varphi_1(x_1, x_2);$$

从式(4 − 32)的第二个方程中虚拟解出 x_2:

$$x_2 = 0.1x_1 + 0.1x_2^2 + 2 \xlongequal{\text{令}} \varphi_2(x_1, x_2).$$

又令
$$\boldsymbol{\Phi}(\boldsymbol{x}) = (\varphi_1(\boldsymbol{x}), \varphi_2(\boldsymbol{x}))^{\mathrm{T}},$$

于是得到虚拟解出的方程组

$$\boldsymbol{x} = \boldsymbol{\Phi}(\boldsymbol{x}),$$

而方程组 $\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{0}$ 与方程组 $\boldsymbol{x} = \boldsymbol{\Phi}(\boldsymbol{x})$ 两者是等价的.

现由方程组 $\boldsymbol{x} = \boldsymbol{\Phi}(\boldsymbol{x})$ 构造迭代式:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{\Phi}(\boldsymbol{x}^{(k)}),$$

其中
$$\boldsymbol{x}^{(k)} = (x_1^{(k)}, x_2^{(k)})^{\mathrm{T}}, \boldsymbol{x}^{(k+1)} = (x_1^{(k+1)}, x_2^{(k+1)}),$$

即
$$\begin{cases} x_1^{(k+1)} = 0.1(x_1^{(k)})^2 + 0.1x_2^{(k)} + 1 \\ x_2^{(k+1)} = 0.1x_1^{(k)} + 0.1(x_2^{(k)})^2 + 2 \end{cases}. \tag{4-33}$$

在迭代式(4 − 33)中,若给定初始值 $\boldsymbol{x}^{(0)} = (x_1^{(0)}, x_2^{(0)})^{\mathrm{T}}$,通过反复代入(4 − 33),可得一系列点(二维向量)

$$\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(k)}, \cdots$$

若该向量序列 $\{\boldsymbol{x}^{(k)}\}$ 存在极限,即

$$\boldsymbol{x}^{(k)} \to \boldsymbol{x}^{(*)} \ (k \to \infty),$$

则由
$$\lim_{k\to\infty} \boldsymbol{x}^{(k+1)} = \boldsymbol{\Phi}(\lim_{k\to\infty} \boldsymbol{x}^{(k)}),$$

即
$$\boldsymbol{x}^* = \boldsymbol{\Phi}(\boldsymbol{x}^*).$$

于是有 $\boldsymbol{F}(\boldsymbol{x}^*) = \boldsymbol{0}$,也就是说 $\boldsymbol{x}^* = (x_1^*, x_2^*)^{\mathrm{T}}$ 是原方程组(4 − 32)的解.

给定初值 $\boldsymbol{x}^{(0)} = (0, 0)^{\mathrm{T}}$,按迭代式(4 − 33)进行计算,其计算结果列于表 4 − 5.

表 4 − 5

k		0	1	2	3	4	5	6	7
$\boldsymbol{x}^{(k)}$	$x_1^{(k)}$	0	1	1.3	1.419	1.477	1.508	1.526	1.537
	$x_2^{(k)}$	0	2	2.5	2.755	2.901	2.989	3.044	3.079

迭代终止法则:

$$\begin{aligned} \|\boldsymbol{x}^{(7)} - \boldsymbol{x}^{(6)}\|_\infty &= \max\{|x_1^{(7)} - x_1^{(6)}|, |x_2^{(7)} - x_2^{(6)}|\} \\ &= \max\{0.011, 0.035\} \\ &= 0.035 = 0.35 \times 10^{-1} < \frac{1}{2} \times 10^{-1}. \end{aligned}$$

取
$$\boldsymbol{x}^* \approx \boldsymbol{x}^{(7)} = (1.537, 3.079)^{\mathrm{T}} \approx (1.5, 3.1)^{\mathrm{T}},$$

则二维向量$(1.5, 3.1)^{\mathrm{T}}$ 即为方程组(4 − 32)的近似解,解向量的每一个分量精确至十分位。

例 11　在化工绝热闪蒸器数学模型中,其液气流量比 α 与物料的平衡常数 k_i 和摩尔分数 z_i 的关系式如下:

$$\alpha = -k_3 - \frac{z_3(1-k_3)}{\frac{z_1(1-k_1)}{\alpha^{(i)}+k_1}+\frac{z_2(1-k_2)}{\alpha^{(i)}+k_2}+\frac{z_4(1-k_4)}{\alpha^{(i)}+k_4}+\frac{z_5(1-k_5)}{\alpha^{(i)}+k_5}}.$$

式中 α 为未知量，$k_i(i=1,2,3,4,5)$，$z_i(i=1,2,3,4,5)$ 为已知量(见平衡常数表 4－6)；欲求 α 满足一定精度的迭代值.

表 4－6

i	1	2	3	4	5
z_i(摩尔分数)	0.220	0.660	0.114	0.002	0.004
k_i(平衡常数)	66.67	50.00	0.015	100.0	33.33

解　简单迭代式为

$$\alpha^{(i+1)} = -k_3 - \frac{z_3(1-k_3)}{\frac{z_1(1-k_1)}{\alpha^{(i)}+k_1}+\frac{z_2(1-k_2)}{\alpha^{(i)}+k_2}+\frac{z_4(1-k_4)}{\alpha^{(i)}+k_4}+\frac{z_5(1-k_5)}{\alpha^{(i)}+k_5}}.$$

给定迭代初值 $\alpha^{(0)}=1$ 及 z_i，k_i 的值：

$$\begin{cases} z_1 = 0.220, k_1 = 66.67 \\ z_2 = 0.660, k_2 = 50.00 \\ z_3 = 0.114, k_3 = 0.015 \\ z_4 = 0.002, k_4 = 100.0 \\ z_5 = 0.004, k_5 = 33.33 \end{cases}.$$

经 3 次迭代得

$$\alpha^{(3)} = 0.114441,$$

$$|\alpha^{(3)} - \alpha^*| = 10^{-6} < \frac{1}{2}\times 10^{-4}.$$

取　　$\alpha^{(*)} \approx \alpha^{(3)} = 0.114441 \approx 0.1144$(精确至第 4 位小数).

4.5.2　Newton－Raphson 迭代法(N－R 迭代法)

设非线性方程组 $\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{0}$，其中

$$\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^{\mathrm{T}},$$

$$\boldsymbol{F}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_n(\boldsymbol{x}))^{\mathrm{T}}.$$

将非线性向量值函数 $\boldsymbol{F}(\boldsymbol{x})$ 线性化，即在 $\boldsymbol{x}^{(0)}$ 处展开，有

$$\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{F}(\boldsymbol{x}^{(0)} + (\boldsymbol{x} - \boldsymbol{x}^{(0)})) \approx \boldsymbol{F}(\boldsymbol{x}^{(0)}) + \frac{\partial \boldsymbol{F}(\boldsymbol{x}^{(0)})}{\partial \boldsymbol{x}}(\boldsymbol{x} - \boldsymbol{x}^{(0)}), \qquad (4-34)$$

其中 $\boldsymbol{F}(\boldsymbol{x})$ 在 $\boldsymbol{x}^{(0)}$ 处的 Jacobi 矩阵为

$$\frac{\partial \boldsymbol{F}(\boldsymbol{x}^{(0)})}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial f_1(\boldsymbol{x})}{\partial x_1} & \frac{\partial f_1(\boldsymbol{x})}{\partial x_2} & \cdots & \frac{\partial f_1(\boldsymbol{x})}{\partial x_n} \\ \frac{\partial f_2(\boldsymbol{x})}{\partial x_1} & \frac{\partial f_2(\boldsymbol{x})}{\partial x_2} & \cdots & \frac{\partial f_2(\boldsymbol{x})}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_n(\boldsymbol{x})}{\partial x_1} & \frac{\partial f_n(\boldsymbol{x})}{\partial x_2} & \cdots & \frac{\partial f_n(\boldsymbol{x})}{\partial x_n} \end{bmatrix}_{\boldsymbol{x}=\boldsymbol{x}^{(0)}}$$

由 $F(x)=0$，据(4－34)有

$$F(x^{(0)})+\frac{\partial F(x^{(0)})}{\partial x}(x-x^{(0)})=0; \qquad (4-35)$$

再由(4－35)解出 x 有

$$x=x^{(0)}-\left[\frac{\partial F(x^{(0)})}{\partial x}\right]^{-1}F(x^{(0)})\xlongequal{\text{记}}x^{(1)},$$

即有

$$x^{(1)}=x^{(0)}-\left[\frac{\partial F(x^{(0)})}{\partial x}\right]^{-1}F(x^{(0)}).$$

若以 $x^{(k)}$ 为初值展开 $F(x)$，类似以上方法得

$$x^{(k+1)}=x^{(k)}-\left[\frac{\partial F(x^{(k)})}{\partial x}\right]^{-1}F(x^{(k)}). \qquad (4-36)$$

称(4－36)为N－R迭代式.

例12　用N－R迭代法解例10中的方程组(4－32)，要求解向量各分量精确至十分位.

解

$$f_1(x)=0.1x_1^2-x_1+0.1x_2+1,$$
$$f_2(x)=0.1x_1-x_2+0.1x_2^2+2.$$

取初值

$$x^{(0)}=(x_1^{(0)},x_2^{(0)})^{\mathrm{T}}=(0,0)^{\mathrm{T}},$$

$$f_1(x^{(0)})=f_1(0,0)=1,\quad f_2(x^{(0)})=f_2(0,0)=2,$$

$$F(x^{(0)})=(f_1(x^{(0)}),f_2(x^{(0)}))^{\mathrm{T}}=(1,2)^{\mathrm{T}},$$

$$\|F(x^{(0)})\|_2=\sqrt{1+4}=2.236,$$

$$\frac{\partial F(x^{(0)})}{\partial x}=\begin{bmatrix}\dfrac{\partial f_1(x)}{\partial x_1} & \dfrac{\partial f_1(x)}{\partial x_2}\\ \dfrac{\partial f_2(x)}{\partial x_1} & \dfrac{\partial f_2(x)}{\partial x_2}\end{bmatrix}_{x=x^{(0)}}$$

$$=\begin{bmatrix}0.2x_1-1 & 0.1\\ 0.1 & 0.2x_2-1\end{bmatrix}_{x^{(0)}}=\begin{pmatrix}-1 & 0.1\\ 0.1 & -1\end{pmatrix}.$$

第一次迭代式为：

$$x^{(1)}=x^{(0)}-\left[\frac{\partial F(x^{(0)})}{\partial x}\right]^{-1}F(x^{(0)}).$$

将上式转化为线性方程组

$$\frac{\partial F(x^{(0)})}{\partial x}(x^{(1)}-x^{(0)})=-F(x^{(0)}),$$

即

$$\begin{pmatrix}-1 & 0.1\\ 0.1 & -1\end{pmatrix}(x^{(1)}-x^{(0)})=-\begin{pmatrix}1\\2\end{pmatrix},$$

$$\begin{pmatrix}-1 & 0.1\\ 0.1 & -1\end{pmatrix}\begin{bmatrix}x_1^{(1)}-x_1^{(0)}\\ x_2^{(1)}-x_2^{(0)}\end{bmatrix}=\begin{pmatrix}-1\\-2\end{pmatrix},\begin{pmatrix}-1 & 0.1\\ 0.1 & -1\end{pmatrix}\begin{bmatrix}x_1^{(1)}\\ x_2^{(1)}\end{bmatrix}=\begin{pmatrix}-1\\-2\end{pmatrix},$$

也即

$$\begin{cases}-x_1^{(1)}+0.1x_2^{(1)}=-1\\ 0.1x_1^{(1)}-x_2^{(1)}=-2\end{cases}.$$

解方程组得 $x_1^{(1)}=1.212,x_2^{(1)}=2.121$ 即

$$x^{(1)}=(x_1^{(1)},x_2^{(1)})^T=(1.212,2.121)^T.$$

第一次迭代结果　　$x^{(1)}=(1.212,2.121)^T$.

因为
$$\begin{aligned}\|x^{(1)}-x^{(0)}\|_\infty &= \max\{|x_1^{(1)}-x_1^{(0)}|,|x_2^{(1)}-x_2^{(0)}|\}\\ &=\max\{1.212,2.121\}\\ &=2.12,\end{aligned}$$

即
$$\|x^{(1)}-x^{(0)}\|_\infty>\frac{1}{2}\times10^{-1},$$

故应继续迭代下去.

请读者通过计算进行比较：本题中的方程组若用简单迭代法迭代 7 次可达到精度十分位(见例 10)，而用 N－R 迭代法需要多少次迭代才能达到同样的精度？

4.5.3　拟 Newton 迭代法(Broyden 方法)

德国数学家 Broyden 对 N－R 迭代法作如下修正，称为修正的 N－R 方法。

(1) 增加迭代步长因子 ω(松弛因子).

$$x^{(k+1)}=x^{(k)}-\omega\left[\frac{\partial F(x^{(k)})}{\partial x}\right]^{-1}F(x^{(k)}),$$

其中
$$\omega=\begin{cases}1 & ,当\|F(x^{(k+1)})\|_2<\|F(x^{(k)})\|_2\\ \dfrac{\sqrt{1+6\eta}-1}{3\eta} & ,当\|F(x^{(k+1)})\|_2\geqslant\|F(x^{(k)})\|_2\end{cases}.$$

而 $\eta=\left[\dfrac{\|F(x^{(k+1)})\|_2}{\|F(x^k)\|_2}\right]^2$.

(2) 构造 Jacobi 矩阵的逆阵的递推关系.

令　　$-\left[\dfrac{\partial F(x^{(k)})}{\partial x}\right]^{-1}=H^{(k)}$,

则　　$H^{(k+1)}=\left[\dfrac{\partial F(x^{(k+1)})}{\partial x}\right]^{-1}=H^{(k)}-\Delta H^{(k)}$.

其中
$$\begin{cases}\Delta H^{(k)}=\dfrac{(H^{(k)}y^{(k)}+\omega\Delta x^{(k)})\Delta x^{(k)^T}H^{(k)}}{\Delta x^{(k)^T}H^{(k)}y^{(k)}}\\ y^{(k)}=F(x^{(k+1)})-F(x^{(k)})\\ \Delta x^{(k)}=H^{(k)}F(x^{(k)})\end{cases}.$$

初值
$$H^{(0)}=\begin{cases}-\left[\dfrac{\partial F(x^{(0)})}{\partial x}\right]^{-1} & (收敛快，稳定性好，计算量大)\\ -I_0 & (I_0\ 单位矩阵，其计算量小但收敛慢)\end{cases}.$$

例 13　解非线性方程组

$$\begin{cases}f_1(x,y)=x^2-xy^2-2=0\\ f_2(x,y)=2x^2-3xy^2+3=0\end{cases}\tag{4-37}$$

用 Broyden 迭代法迭代两步. 取初值 $x^{(0)}=(x_0,y_0)^T=(1,1)^T$.

解　令　$x=(x,y)^T$,

$f_1(x)=x^2-xy^2-2$,

$$f_2(x) = 2x^2 - 3xy^2 + 3,$$
$$F(x) = (f_1(x), f_2(x))^T,$$
则方程组(4－37)为
$$F(x) = 0.$$

第一次迭代：

① 计算： $F(x^{(0)}) = (f_1(x^{(0)}), f_2(x^{(0)})^T,$

$$f_1(x^{(0)}) = f_1(x_0, y_0) = f_1(1,1) = -2,$$
$$f_2(x^{(0)}) = f_2(x_0, y_0) = f_2(1,1) = 2,$$
$$F(x^{(0)}) = (-2,2)^T,$$
$$\|F(x^{(0)})\|_2 = \sqrt{(-2)^2 + 2^2} = 2.83.$$

② 计算 $H^{(0)} = -\dfrac{\partial F(x^{(0)})}{\partial x}$；

若用解析法计算，则有

$$\frac{\partial F(x^{(0)})}{\partial x} = \begin{pmatrix} 2x - y^2 & -2xy \\ 4x - 3y^2 & -6xy \end{pmatrix}_{x^{(0)}} = \begin{pmatrix} 1 & -2 \\ 1 & -6 \end{pmatrix};$$

现用差分法计算(取差分步长 $h = 0.001$)：

因为
$$\frac{\partial f_1}{\partial x} = \frac{f_1(1.001,1) - f_1(1,1)}{0.001} = 0.001,$$
$$\frac{\partial f_1}{\partial y} = \frac{f_1(1,1.001) - f_1(1,1)}{0.001} = -2.001,$$
$$\frac{\partial f_2}{\partial x} = \frac{f_2(1.001,1) - f_2(1,1)}{0.001} = 1.002,$$
$$\frac{\partial f_2}{\partial y} = \frac{f_2(1,1.001) - f_2(1,1)}{0.001} = -6.003,$$

所以
$$\frac{\partial F(x^{(0)})}{\partial x} = \begin{pmatrix} 1.001 & -2.001 \\ 1.002 & -6.003 \end{pmatrix}.$$

故取 $$H^{(0)} = -\left[\frac{\partial F(x^{(0)})}{\partial x}\right]^{-1} = \begin{pmatrix} -1.499 & 0.500 \\ -0.250 & 0.250 \end{pmatrix}.$$

③ 计算 $\Delta x^{(0)}$：
$$\Delta x^{(0)} = H^{(0)} F(x^{(0)}) = (4.00, 1.00)^T.$$

④ 取 $\omega = 1$，计算 $x^{(1)}$：
$$x^{(1)} = x^{(0)} + \Delta x^{(0)} = (5.00, 2.00)^T,$$

于是有
$$f_1(x^{(1)}) = f_1(4.00, 1.00) = 2.98,$$
$$f_2(x^{(1)}) = f_2(4.00, 1.00) = -7.05,$$
$$F(x^{(1)}) = (2.98, -7.05)^T,$$
$$\|F(x^{(1)})\|_2 = \sqrt{2.98^2 + (-7.05)^2} = 7.65.$$

因为 $\|F(x^{(1)})\|_2 > \|F(x^{(0)})\|_2$，故需调整 ω 的值.

⑤ 修正 ω：

$$\eta=(\|F(x^{(1)})\|_2/\|F(x^{(0)})\|_2)^2=\left(\frac{7.65}{2.83}\right)^2=7.31,$$

$$\omega=\frac{\sqrt{1+6\eta}-1}{3\eta}=0.260,$$

$$x^{(1)}=x^{(0)}+\omega\Delta x^{(0)}=\begin{pmatrix}1\\1\end{pmatrix}+0.260\begin{pmatrix}4.00\\1.00\end{pmatrix}=\begin{pmatrix}2.04\\1.26\end{pmatrix},$$

$$f_1(x^{(1)})=f_1(2.04,1.26)=-1.08,$$

$$f_2(x^{(1)})=f_2(2.04,1.26)=1.60,$$

$$F(x^{(1)})=(-1.08,1.60)^{\mathrm{T}},$$

$$\|F(x^{(1)})\|_2=\sqrt{(-1.08)^2+1.60^2}=1.93.$$

因为$\|F(x^{(1)})\|_2<\|F(x^{(0)})\|_2$，因此第一次迭代结束，即

$$x^{(1)}=(2.04,1.26)^{\mathrm{T}}.$$

第二次迭代：

① 计算 $H^{(1)}=-\left[\frac{\partial F(x^{(1)})}{\partial x}\right]^{-1}$，采用 Broyden 矩阵迭代式

$$H^{(1)}=H^{(0)}+\Delta H^{(0)}.$$

而计算 $\Delta H^{(0)}$ 需作如下计算：

$$y^{(0)}=F(x^{(1)})-F(x^{(0)})=\begin{pmatrix}-1.08\\1.60\end{pmatrix}-\begin{pmatrix}-2\\2\end{pmatrix}=\begin{pmatrix}0.92\\-0.40\end{pmatrix},$$

$$H^{(0)}y^{(0)}=\begin{pmatrix}-1.499&0.500\\-0.250&0.250\end{pmatrix}\begin{pmatrix}0.92\\-0.40\end{pmatrix}=\begin{pmatrix}-1.58\\-0.329\end{pmatrix},$$

$$\Delta x^{(0)\mathrm{T}}H^{(0)}=(4.00,1.00)\begin{pmatrix}-1.499&0.500\\-0.250&0.250\end{pmatrix}=(-6.24,2.25),$$

$$\Delta x^{(0)\mathrm{T}}H^{(0)}y^{(0)}=(-6.24,2.25)\begin{pmatrix}0.92\\-0.40\end{pmatrix}=-6.63,$$

$$H^{(0)}y^{(0)}+\omega\Delta x^{(0)}=\begin{pmatrix}-1.58\\-0.329\end{pmatrix}+0.260\begin{pmatrix}4.00\\1.00\end{pmatrix}=\begin{pmatrix}-0.54\\-0.069\end{pmatrix},$$

$$\Delta H^{(0)}=-\frac{(H^{(0)}y^{(0)}+\omega\Delta x^{(0)})\Delta x^{(0)\mathrm{T}}H^{(0)}}{\Delta x^{(0)\mathrm{T}}H^{(0)}y^{(0)}}=\begin{pmatrix}-0.508&0.183\\-0.065&0.023\end{pmatrix},$$

$$H^{(1)}=H^{(0)}+\Delta H^{(0)}=\begin{pmatrix}-2.007&0.683\\-0.315&0.273\end{pmatrix}.$$

② 计算：

$$\Delta x^{(1)}=H^{(1)}F(x^{(1)})=\begin{pmatrix}-2.007&0.683\\-0.315&0.273\end{pmatrix}\begin{pmatrix}-1.08\\1.60\end{pmatrix}=\begin{pmatrix}3.260\\0.777\end{pmatrix},$$

③ 取 $\omega=1$，计算 $x^{(2)}$：

$$x^{(2)}=x^{(1)}+\Delta x^{(1)}=\begin{pmatrix}2.04\\1.26\end{pmatrix}+\begin{pmatrix}3.260\\0.77\end{pmatrix}=\begin{pmatrix}5.300\\2.037\end{pmatrix},$$

$f_1(\boldsymbol{x}^{(2)}) = f_1(5.300, 2.037) = 4.098,$
$f_2(\boldsymbol{x}^{(2)}) = f_2(5.300, 2.037) = -6.795,$
$\boldsymbol{F}(\boldsymbol{x}^{(2)}) = (4.098, -6.795)^{\mathrm{T}},$
$\|\boldsymbol{F}(\boldsymbol{x}^{(2)})\|_2 = 7.935.$
因为$\|\boldsymbol{F}(\boldsymbol{x}^{(2)})\|_2 > \|\boldsymbol{F}(\boldsymbol{x}^{(1)})\|_2 = 1.93$,因此须修改 ω 继续计算 $x^{(2)}$.
④ 修改 ω:

$$\eta = \left(\frac{\|\boldsymbol{F}(\boldsymbol{x}^{(2)})\|_2}{\|\boldsymbol{F}(\boldsymbol{x}^{(1)})\|_2}\right)^2 = \left(\frac{7.935}{1.93}\right)^2 = 16.90,$$

$$\omega = \frac{\sqrt{1+6\eta}-1}{3\eta} = 0.17,$$

$$\boldsymbol{x}^{(2)} = \boldsymbol{x}^{(1)} + \omega\Delta\boldsymbol{x}^{(1)} = \begin{pmatrix}2.04\\1.26\end{pmatrix} + 0.17\begin{pmatrix}3.260\\0.777\end{pmatrix} = \begin{pmatrix}2.594\\1.479\end{pmatrix},$$

$f_1(\boldsymbol{x}^{(2)}) = f_1(2.594, 1.479) = -0.95,$
$f_2(\boldsymbol{x}^{(2)}) = f_2(2.594, 1.479) = -0.56,$
$\boldsymbol{F}(\boldsymbol{x}^{(2)}) = (-0.95, -0.56)^{\mathrm{T}},$
$\|\boldsymbol{F}(x^{(2)})\|_2 = 1.103.$
因为$\|\boldsymbol{F}(\boldsymbol{x}^{(2)})\|_2 < \|\boldsymbol{F}(\boldsymbol{x}^{(1)})\|_2 = 1.93$,所以第二次迭代结束.

习题 4

*1. 用 Jacobi 迭代法和 Seidel 迭代法求解下列方程组(要求解向量的各分量准确至三位小数).

(1) $\begin{bmatrix}2&1&0\\1&2&1\\0&1&2\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix} = \begin{bmatrix}3\\-5\\4\end{bmatrix}$;　　(2) $\begin{bmatrix}-8&1&1\\1&-5&1\\1&1&-4\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix} = \begin{bmatrix}1\\16\\7\end{bmatrix}$.

*2. 以下列矩阵为系数矩阵构成线性方程组. 试判断它们对 Jacobi 迭代法,Seidel 迭代法的收敛性.

(1) $\begin{bmatrix}5&2&1\\1&3&2\\1&1&2\end{bmatrix}$;　　(2) $\begin{pmatrix}1&2\\3&2\end{pmatrix}$.

*3. 设方程组

$$\begin{pmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{pmatrix}\begin{pmatrix}x_1\\x_2\end{pmatrix} = \begin{pmatrix}b_1\\b_2\end{pmatrix},\quad a_{11}\neq 0,\quad a_{22}\neq 0,$$

试证明用 Jacobi 迭代法收敛的充要条件是

$$r = \left|\frac{a_{12}a_{21}}{a_{11}a_{22}}\right| < 1.$$

*4. 设 $\boldsymbol{A} = \begin{bmatrix}1&a&a\\a&1&a\\a&a&1\end{bmatrix}$,$a$ 为实数,求:

(1)若 A 正定，a 的取值范围；

(2)若简单迭代法收敛，a 的取值范围.

*5. 求下列矩阵的 $\|A\|_1$，$\|A\|_2$，$\|A\|_\infty$，$\rho(A)$.

(1)$A=\begin{bmatrix}0.6 & 0.5\\0.1 & 0.3\end{bmatrix}$；　　(2)$A=\begin{bmatrix}5 & 7 & 3\\7 & 11 & 2\\3 & 2 & 6\end{bmatrix}$.

6. 证明向量范数有下列等价性质.

(1) $\|x\|_2\leqslant\|x\|_1\leqslant\sqrt{n}\|x\|_2$；

(2) $\|x\|_\infty\leqslant\|x\|_1\leqslant n\|x\|_\infty$；

(3) $\|x\|_\infty\leqslant\|x\|_2\leqslant\sqrt{n}\|x\|_\infty$.

7. 用 SOR 方法解方程组(取 $\omega=0.9$)

$$\begin{cases}5x_1+2x_2+x_3=-12,\\-x_1+4x_2+2x_3=20,\\2x_1-3x_2+10x_3=3.\end{cases}$$

要求当 $\|x^{(k+1)}-x^{(k)}\|_\infty<\frac{1}{2}\times10^{-3}$时，迭代终止.

8. 用 Newton--Raphson 迭代法求解例 12 中的方程组，要求解向量各分量精确至十分位(给定初始向量 $x^{(0)}=(1.212,2.121)^T$).

9. 用 Broyden 迭代法解例 13 中的方程组，迭代一次. 给定初始向量 $x^{(0)}=(5.300,2.037)^T$.

注　打 * 号的习题在本书附录Ⅱ作了详解.

*第 5 章　矩阵的特征值和特征向量的计算

5.1　预备知识(矩阵的特征值和特征向量)

定义 5.1　设 $\boldsymbol{A}=(a_{ij})_{n\times n}, a_{ij}\in R$,若对于 $\forall \boldsymbol{x}\neq\boldsymbol{0}(\boldsymbol{x}\in R^n$ 或者 $\boldsymbol{x}\in C^n)$及 $\forall\lambda\in R$(或 $\in C$)有下式成立

$$\boldsymbol{Ax}=\lambda\boldsymbol{x}, \tag{5-1}$$

则称 λ 为 $\boldsymbol{A}$ 的对应于 $\boldsymbol{x}$ 的特征值;$\boldsymbol{x}$ 为 $\boldsymbol{A}$ 的属于 λ 的特征向量.

将(5-1)式整理成

$$(\boldsymbol{A}-\lambda\boldsymbol{I})\boldsymbol{x}=\boldsymbol{0}, \tag{5-2}$$

则式(5-2)是一个齐次线性方程组. 因为 $\boldsymbol{x}\neq\boldsymbol{0}$,故有

$$\det(\boldsymbol{A}-\lambda\boldsymbol{I})=0, \tag{5-3}$$

即

$$\begin{vmatrix} a_{11}-\lambda & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22}-\lambda & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn}-\lambda \end{vmatrix}=0,$$

亦即

$$\lambda^n-p_1\lambda^{n-1}+p_2\lambda^{n-2}+\cdots+(-1)^n p_n=0.$$

我们称式(5-3)为 $\boldsymbol{A}$ 的特征方程,它是一个含 λ 的 n 次代数方程,该方程的根即是矩阵 $\boldsymbol{A}$ 的特征值.

矩阵 $\boldsymbol{A}$ 的特征值和相应的特征向量具有如下的一些性质:

(1)n 阶方阵 $\boldsymbol{A}$ 有 n 个特征值 $\lambda_i(i=1,2,\cdots,n)$,即特征方程 $\det(\boldsymbol{A}-\lambda\boldsymbol{I})=0$ 有 n 个根,且

$$\sum_{i=1}^{n}\lambda_i=p_1=\sum_{i=1}^{n}a_{ii}\overset{\text{记}}{=}\mathrm{tr}(\boldsymbol{A}),$$

$$\prod_{i=1}^{n}\lambda_i=(-1)^n p_n=\det(\boldsymbol{A}).$$

(2)特征方程$(\boldsymbol{A}-\lambda\boldsymbol{I})\boldsymbol{x}=\boldsymbol{0}$ 的每一个根 λ 对应无穷多个属于 λ 的特征向量 $\boldsymbol{x}\neq\boldsymbol{0}$;而每一个特征向量 $\boldsymbol{x}\neq\boldsymbol{0}$ 仅对应一个特征值 λ.

(3)特征向量的线性性质. 设 $\boldsymbol{x}\neq\boldsymbol{0}$ 满足

$$\boldsymbol{Ax}=\lambda\boldsymbol{x},$$

则 $k\boldsymbol{x}\neq\boldsymbol{0}$ 满足

$$\boldsymbol{A}(k\boldsymbol{x})=\lambda(k\boldsymbol{x})\quad(k\text{ 为任意实数}).$$

一般地,设 $\boldsymbol{x}_i\neq 0(i=1,2,\cdots,m)$满足

$$\boldsymbol{Ax}_i=\lambda\boldsymbol{x}_i,$$

则 $\sum_{i=1}^{m} k_i x_i$ 满足

$$A(\sum_{i=1}^{m} k_i x_i) = \lambda(\sum_{i=1}^{m} k_i x_i),$$

其中 $k_i (i=1,2,\cdots,m)$ 为常数.

(4)设 $\lambda, x \neq 0$ 分别为 A 的特征值和属于该特征值的特征向量,则

$$(kA)x=(k\lambda)x \quad (k \text{ 为任意实数}),$$

$$(A-\mu I)x=(\lambda-\mu)x \quad (\mu \text{ 为任意实数}),$$

$$A^{-1}x=\frac{1}{\lambda}x,$$

$$(A-\mu I)^{-1}x=\frac{1}{\lambda-\mu}x.$$

5.2 幂法与反幂法

5.2.1 基本幂法

定义 5.1(主特征值为单实根的情形) 设 $A=(a_{ij})_{n\times n}, a_{ij}\in R$,且

(1)A 有完全的特征向量系,即 A 有 n 个线性无关的特征向量 $x_1, x_2, \cdots, x_n$;

(2)相应的特征值 $\lambda_i (i=1,2,\cdots,n)$ 满足

$$|\lambda_1|>|\lambda_2|\geqslant|\lambda_3|\geqslant\cdots\geqslant|\lambda_n|.$$

对 A 作幂法,即任取非零向量 v_0,并令

$$v_k=Av_{k-1}, \quad k=0,1,2,\cdots$$

得向量序列 $\{v_k\}_{k=0,1,2,\cdots}$,则 $\{v_k\}$ 满足

(1)$\lim\limits_{k\to\infty}\frac{v_k}{\lambda_1^k}=\alpha_1 x_1$,即当 k 充分大时,

$$v_k \approx \alpha_1 \lambda_1^k x_1. \tag{5-4}$$

其中 x_1 是 A 的属于主特征值 λ_1 的主特征向量;$\alpha_1\lambda_1^k x_1$ 仍是 A 的属于主特征值 λ_1 的主特征向量.

(2)$\lim\limits_{k\to\infty}\frac{(v_{k+1})_i}{(v_k)_i}=\lambda_1, i=1,2,\cdots,n$,即当 k 充分大时,

$$\frac{(v_{k+1})_i}{(v_k)_i}\approx\lambda_1, \quad i=1,2,\cdots,n. \tag{5-5}$$

证 因为 $x_1, x_2, \cdots, x_n$ 线性无关,故任意非零向量 v_0 可由它们线性表出:

$$\begin{aligned} v_0 &= \alpha_1 x_1+\alpha_2 x_2+\cdots+\alpha_n x_n, \\ v_k &= Av_{k-1}=A^2 v_{k-2}=\cdots=A^k v_0 \\ &= \alpha_1 A^k x_1+\alpha_2 A^k x_2+\cdots+\alpha_n A^k x_n \\ &= \alpha_1\lambda_1^k x_1+\alpha_2\lambda_2^k x_2+\cdots+\alpha_n\lambda_n^k x_n \\ &= \lambda_1^k\left[\alpha_1 x_1+\alpha_2\left(\frac{\lambda_2}{\lambda_1}\right)^k x_2+\cdots+\alpha_n\left(\frac{\lambda_n}{\lambda_1}\right)^k x_n\right]. \end{aligned}$$

设 $\alpha_1 \neq 0$，当 k 充分大时，因为

$$\left|\frac{\lambda_2}{\lambda_1}\right|<1, \frac{\lambda_3}{\lambda_1}<1, \cdots, \left|\frac{\lambda_n}{\lambda_1}\right|<1,$$

所以

$$v_k \approx \alpha_1 \lambda_1^k x_1.$$

据特征向量的线性性质，v_k 可近似作为属于 λ_1 的特征向量.

又因为当 k 充分大时

$$v_k \approx \alpha_1 \lambda_1^k x_1, \quad v_{k+1} \approx \alpha_1 \lambda_1^{k+1} x_1,$$

所以

$$\frac{(v_{k+1})_i}{(v_k)_i} \approx \frac{\alpha_1 \lambda_1^{k+1} x_{1i}}{\alpha_1 \lambda_1^k x_{1i}} = \lambda_1, \quad i=1,2,\cdots,n.$$

定理 5.2（主特征值为重实根的情形） 设 $\boldsymbol{A}=(a_{ij})_{n\times n}, a_{ij} \in R$，且

(1) $\boldsymbol{A}$ 有完全的特征向量系；

(2) $\boldsymbol{A}$ 的主特征值 λ_1 为 r 重根，即

$$\lambda_1 = \lambda_2 = \cdots = \lambda_r,$$

$$|\lambda_r| > |\lambda_{r+1}| \geqslant \cdots \geqslant |\lambda_n|.$$

对 $\boldsymbol{A}$ 作幂法，即任取非零向量 v_0，并令

$$v_k = \boldsymbol{A} v_{k-1}, \quad k=1,2,\cdots$$

得向量系 $\{v_k\}_{k=0,1,\cdots}$，则 $\{v_k\}$ 满足

(1) $\lim\limits_{k\to\infty} \frac{v_k}{\lambda_1^k} = \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_r x_r$； (5-6)

(2) $\lim\limits_{k\to\infty} \frac{(v_{k+1})_i}{(v_k)_i} = \lambda_1.$ (5-7)

定理 5.2 的证明与定理 5.1 基本相同，建议读者自己练习.

基本幂法在实际计算时，由于迭代过程中 v_k 的模可能过大或过小，从而产生计算机运算的溢出. 为防止此种情形的发生，通常采用规范化措施.

5.2.2 规范化幂法

据定理 5.1 及定理 5.2，当 k 充分大时有

$$v_k \approx \alpha_1 \lambda_1^k x_1,$$

或

$$v_k = \lambda_1^k (\alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_r x_r),$$

且

$$\frac{(v_{k+1})_i}{(v_k)_i} \approx \lambda_1, \quad i=1,2,\cdots,n.$$

当 $|\lambda_1|<1$ 时，$(v_k)_i \to 0 (k\to\infty)$，此时计算 λ_1 会产生溢出；当 $|\lambda_1|>1$ 时 $(v_k)_i \to \infty (k\to\infty)$，此时计算 v_k 也会产生溢出. 为此需对迭代向量 v_k 规范化.

设

$$v = (v_1, v_2, \cdots, v_n)^{\mathrm{T}},$$

记 $\max(v) = v_{i_0}$，且满足

$$|v_{i_0}| = \max_{1\leqslant i\leqslant n} |v_i| = \|v\|_\infty.$$

作

$$u = v/\max(v) = [v_1/\max(v), \cdots, 1^{(i_0)}, \cdots, v_n/\max(v)]^{\mathrm{T}},$$

规范化向量 u 的第 i_0 个分量为 1（即其最大分量为 1）. 由此得到规范化幂法的计算公式：

给定 $v_0 = u_0 \neq \boldsymbol{0}$，令

$$\begin{cases} v_k = Au_{k-1} & \text{(迭代向量)} \\ u_k = v_k/\max(v_k) & \text{(规范化迭代向量)} \end{cases}, \quad k=1,2,\cdots$$

且有如下结论.

定理 5.3 设 $A=(a_{ij})_{n\times n}, a_{ij}\in R$,且

(1)A 有完全的特征向量系;

(2)A 的主特征值或为单实根或为重实根.

对 A 作规范化幂法,即对给定 $v_0=u_0\neq 0$,令

$$\begin{cases} v_k = Au_{k-1}, \\ u_k = v_k/\max(v_k), \quad k=1,2,\cdots \end{cases} \tag{5-8}$$

则有

$$\begin{cases} \lim\limits_{k\to\infty} u_k = x_1/\max(x_1), \\ \lim\limits_{k\to\infty} \max(v_k) = \lambda_1. \end{cases} \tag{5-9}$$

规范化幂法的计算步骤如表 5-1 所示.

表 5-1

$v_0=u_0$	$(v_0)_1$	$(v_0)_2$	…	$(v_0)_n$	
$v_1=Au_0$	$(v_1)_1$	$(v_1)_2$	…	$(v_1)_n$	$\max(v_1)$
$u_1=v_1/\max(v_1)$	$(u_1)_1$	$(u_1)_2$	…	$(u_1)_n$	
$v_2=Au_1$	$(v_2)_1$	$(v_2)_2$	…	$(v_2)_n$	$\max(v_2)$
$u_2=v_2/\max(v_2)$	$(u_2)_1$	$(u_2)_2$	…	$(u_2)_n$	
⋮	⋮	⋮		⋮	⋮
$v_{n-1}=Au_{n-2}$	$(v_{n-1})_1$	$(v_{n-1})_2$	…	$(v_{n-1})_n$	$\max(v_{n-1})$
$u_{n-1}=v_{n-1}/\max(v_{n-1})$	$(u_{n-1})_1$	$(u_{n-1})_2$	…	$(u_{n-1})_n$	
$v_n=Au_{n-1}$	$(v_n)_1$	$(v_n)_2$	…	$(v_n)_n$	$\max(v_n)$
$u_n=v_n/\max(v_n)$	$(u_n)_1$	$(u_n)_2$	…	$(u_n)_n$	

算法终止法则:

对于给定的 $\varepsilon>0$,当 $\|u_n-u_{n-1}\|_\infty=\max\limits_{1\leq i\leq n}|(u)_i-(u_{n-1})_i|<\varepsilon$ 时,取

$$x_1\approx u_n \quad \text{(相应的主特征向量)},$$

$$\lambda_1\approx\max(v_n) \quad \text{(主特征值)}.$$

例 1 用规范化幂法计算 A 的主特征值(最大模特征值)及相应的特征向量.式中

$$A=\begin{bmatrix} 1.0 & 1.0 & 0.5 \\ 1.0 & 1.0 & 0.25 \\ 0.5 & 0.25 & 2.0 \end{bmatrix}.$$

解 设 $u_0=v_0=(1,1,1)^T$,则有

$$\begin{cases} v_1=Au_0=(2.50,2.25,2.75)^T, \\ \max(v_1)=2.75, \\ u=v_1/\max(v_1)=(0.909\,1,0.818\,2,1)^T; \end{cases}$$

$$\begin{cases} v_2 = Au_1 = (2.2273, 1.9773, 1.6591)^T, \\ \max(v_2) = 2.2273, \\ u_2 = v_2/\max(v_2) = (1, 0.8878, 0.7449)^T; \end{cases}$$

$$\vdots$$

$$\begin{cases} v_{19} = Au_{18} = (1.8978, 1.6479, 2.5365)^T, \\ \max(v_{19}) = 2.5365, \\ u_{19} = v_{19}/\max(v_{19}) = (0.7482, 0.6497, 1)^T; \end{cases}$$

$$\begin{cases} v_{20} = Au_{19} = (1.8979, 1.6479, 2.5365)^T, \\ \max(v_{20}) = 2.5365, \\ u_{20} = v_{20}/\max(v_{20}) = (0.7482, 0.6497, 1)^T. \end{cases}$$

因为 $\qquad \| u_{20} - u_{19} \|_\infty = \max\{0,0,0\} = 0,$

所以 $\qquad x_1 \approx u_{20} = (0.7482, 0.6497, 1)^T, \quad \lambda_1 \approx \max(v_{20}) = 2.5365.$

5.2.3 原点平移法

用幂法计算主特征值时，据 λ_1 为单实根或 r 重根两种情形，有如下算式成立：

$$v_k = \lambda_1^k \left[\alpha_1 x_1 + \alpha_2 \left(\frac{\lambda_2}{\lambda_1} \right)^k x_2 + \cdots + \alpha_n \left(\frac{\lambda_n}{\lambda_1} \right)^k x_n \right],$$

或 $$v_k = \lambda_1^k \left[\alpha_1 x_1 + \cdots + \alpha_r x_r + \alpha_{r+1} \left(\frac{\lambda_{r+1}}{\lambda_1} \right)^k x_{r+1} + \cdots + \alpha_n \left(\frac{\lambda_n}{\lambda_1} \right)^k x_n \right].$$

在第一种情形中，收敛速度取决于 $\left| \frac{\lambda_2}{\lambda_1} \right|$ 的值；在第二种情形中，收敛速度取决于 $\left| \frac{\lambda_{r+1}}{\lambda_1} \right|$ 的值. 以第一种情形为例，当比值 $\left| \frac{\lambda_2}{\lambda_1} \right|$ 接近于 1 时，即 $|\lambda_1|$ 和 $|\lambda_2|$ 的值很接近，此时 $\alpha_2 \left[\frac{\lambda_2}{\lambda_1} \right]^k x_2$ 这一项不会随 k 增大而迅速趋于零，即是说迭代过程收敛极慢. 一个补救的方法是原点平移法.

设矩阵 $B = A - pI$，p 为选择的平移量；A 的特征值为 $\lambda_i (i=1,2,\cdots,n)$，则 B 的特征值应为 $\lambda_i - p (i=1,2,\cdots,n)$，并且 A，B 的特征向量相同.

例 2 设矩阵 $A = (a_{ij})_{4\times4}$，其 4 个特征值分别为 $\lambda_j = 15 - j (j=1,2,3,4)$，即 $\lambda_1 = 14$，$\lambda_2 = 13$，$\lambda_3 = 12$，$\lambda_4 = 11$，显然 $\left| \frac{\lambda_2}{\lambda_1} \right| = 0.9286$ 与 1 较接近.

现作平移变换，令

$$B = A - pI \quad (p = 12),$$

则 B 的 4 个特征值分别为 $\mu_1 = 2$，$\mu_2 = 1$，$\mu_3 = 0$，$\mu_4 = -1$，故有

$$\left| \frac{\mu_2}{\mu_1} \right| = \left| \frac{\lambda_2 - p}{\lambda_1 - p} \right| = \frac{1}{2} < 0.9286.$$

若用幂法求 B 的主特征值 μ_1 时，决定收敛速度的比值 $\left| \frac{\mu_2}{\mu_1} \right| = 0.5$，这比 $\left| \frac{\lambda_2}{\lambda_1} \right| = 0.9286$ 小得多.

下面的关键在如何选择既能明显地提高收敛速度，又不致于求出的是另一个特征值的方法.

设 A 的特征值满足

$$|\lambda_1|>|\lambda_2|\geqslant|\lambda_3|\geqslant\cdots\geqslant|\lambda_{n-1}|>|\lambda_n|.$$

现令 $B=A-pI$，其特征值为 $\lambda_i-p(i=1,2,\cdots,n)$. 易证：对任意的 p，B 的主特征值或者是 λ_1-p，或者是 λ_n-p. 事实上有以下三种情况成立.

(1)若 $p\approx\lambda_1$，则有

$$|\lambda_1-p|<|\lambda_i-p|<|\lambda_n-p|,\quad i\neq1,n; \tag{5-10}$$

(2)若 $p\approx\lambda_n$，则有

$$|\lambda_1-p|>|\lambda_i-p|>|\lambda_p|,\quad i\neq1,n; \tag{5-11}$$

(3)若 $p\approx\lambda_i(i\neq1,n)$，或者式(5－10)成立，或者式(5－11)成立.

以下给出确定合适的 p 值的两种方案.

(1)若希望计算 λ_1 及相应的 x_1，应选择 p 满足 $|\lambda_1-p|>|\lambda_n-p|$，且使

$$\omega=\max\left\{\left|\frac{\lambda_2-p}{\lambda_1-p}\right|,\left|\frac{\lambda_n-p}{\lambda_1-p}\right|\right\}=\min.$$

易证 $\omega=\min$ 的充分必要条件是

$$\left|\frac{\lambda_2-p}{\lambda_1-p}\right|=\left|\frac{\lambda_n-p}{\lambda_1-p}\right|,$$

即

$$|\lambda_2-p|=|\lambda_n-p|. \tag{5-12}$$

在式(5－12)中，若 $\lambda_2-p=\lambda_n-p$，得 $\lambda_2=\lambda_n$，此时不能确定 p 值；若 $\lambda_2-p=-(\lambda_n-p)$，则得

$$p=\frac{\lambda_2+\lambda_n}{2}. \tag{5-13}$$

(2)若希望计算 λ_n 及相应的 x_n，应选择 p 满足 $|\lambda_n-p|>|\lambda_1-p|$，且使

$$\omega=\max\left\{\left|\frac{\lambda_{n-1}-p}{\lambda_n-p}\right|,\left|\frac{\lambda_1-p}{\lambda_n-p}\right|\right\}=\min.$$

同样易证 $\omega=\min$ 的充分必要条件是

$$\left|\frac{\lambda_{n-1}-p}{\lambda_n-p}\right|=\left|\frac{\lambda_1-p}{\lambda_n-p}\right|,$$

即

$$|\lambda_{n-1}-p|=|\lambda_1-p|. \tag{5-14}$$

此时应取 $\lambda_{n-1}-p=-(\lambda_1-p)$，从而解得

$$p=\frac{(\lambda_1+\lambda_{n-1})}{2}. \tag{5-15}$$

归纳上述讨论，我们得到原点平移法的计算步骤：

设矩阵 $A=(a_{ij})_{n\times n}$，其特征值满足

$$|\lambda_1|>|\lambda_2|\geqslant|\lambda_3|\geqslant\cdots\geqslant|\lambda_{n-1}|>|\lambda_n|.$$

(1)计算 λ_1 及相应的 x_1.

取 $p=\frac{\lambda_2+\lambda_n}{2}$，对 $B=A-pI$ 作规范化幂法，即

$$\begin{cases} \boldsymbol{u}_0 = \boldsymbol{v}_0 \neq \boldsymbol{0}, \\ \boldsymbol{v}_k = (\boldsymbol{A} - p\boldsymbol{I})\boldsymbol{u}_{k-1}, \\ \boldsymbol{u}_k = \boldsymbol{v}_k / \max(\boldsymbol{v}_k), \quad k = 1, 2, \cdots \end{cases} \tag{5-16}$$

则有

$$\lim_{k\to\infty} \boldsymbol{u}_k = \boldsymbol{x}_1 / \max(\boldsymbol{x}_1), \tag{5-17}$$

$$\lim_{k\to\infty} \max(\boldsymbol{v}_k) = \lambda_1 - p. \tag{5-18}$$

即是说规范化向量 $\boldsymbol{u}_k$ 的极限是 $\boldsymbol{A}$ 的对应于 λ_1 的主特征向量；迭代向量 $\boldsymbol{v}_k$ 的最大模分量的极限是 $\boldsymbol{B}$ 的主特征值，从而 $(\lambda_1 - p) + p = \lambda_1$ 即是 $\boldsymbol{A}$ 的主特征值.

(2)计算 λ_n(最小模特征值)及相应的特征向量 $\boldsymbol{x}_n$.

取 $p = (\lambda_1 + \lambda_{n-1})/2$，对 $\boldsymbol{B} = \boldsymbol{A} - p\boldsymbol{I}$ 作幂法，即

$$\begin{cases} \boldsymbol{u}_0 = \boldsymbol{v}_0 \neq \boldsymbol{0}, \\ \boldsymbol{v}_k = (\boldsymbol{A} - p\boldsymbol{I})\boldsymbol{u}_{k-1}, \\ \boldsymbol{u}_k = \boldsymbol{v}_k / \max(\boldsymbol{v}_k), \quad k = 1, 2, \cdots \end{cases} \tag{5-19}$$

则有

$$\lim_{k\to\infty} \boldsymbol{u}_k = \boldsymbol{x}_n / \max(\boldsymbol{x}_n), \tag{5-20}$$

$$\lim_{k\to\infty} \max(\boldsymbol{v}_k) = \lambda_n - p. \tag{5-21}$$

即是说规范化向量 $\boldsymbol{u}_k$ 的极限是 $\boldsymbol{B}$ 的主特征向量；迭代向量 $\boldsymbol{v}_k$ 的最大模分量的极限是 $\boldsymbol{B}$ 的主特征值，从而 $(\lambda_n - p) + p = \lambda_n$ 即是 $\boldsymbol{A}$ 的最小模特征值.

例 3 用原点平移法计算例 1 中矩阵 $\boldsymbol{A}$ 的主特征值及相应的特征向量.

解 取 $p = 0.75$，$\boldsymbol{B} = \boldsymbol{A} - p\boldsymbol{I}$，且

$$\boldsymbol{B} = \begin{bmatrix} 0.25 & 1 & 0.5 \\ 1 & 0.25 & 0.25 \\ 0.5 & 0.25 & 1.25 \end{bmatrix}.$$

对 $\boldsymbol{B}$ 作规范化幂法：

令 $\boldsymbol{u}_0 = \boldsymbol{v}_0 = (1,1,1)^{\mathrm{T}}$，则有

$$\begin{cases} \boldsymbol{v}_1 = \boldsymbol{B}\boldsymbol{u}_0 = (1.75, 1.5, 2)^{\mathrm{T}}, \\ \max(\boldsymbol{v}_1) = 2, \\ \boldsymbol{u}_1 = \boldsymbol{v}_1/\max(\boldsymbol{v}_1) = (0.875, 0.75, 1)^{\mathrm{T}}; \end{cases}$$

$$\begin{cases} \boldsymbol{v}_2 = \boldsymbol{B}\boldsymbol{u}_1 = (1.468\,8, 1.312\,5, 1.875)^{\mathrm{T}}, \\ \max(\boldsymbol{v}_2) = 1.875, \\ \boldsymbol{u}_2 = \boldsymbol{v}_2/\max(\boldsymbol{v}_2) = (0.783\,4, 0.7, 1)^{\mathrm{T}}; \end{cases}$$

$$\vdots$$

$$\begin{cases} \boldsymbol{v}_9 = \boldsymbol{B}\boldsymbol{u}_8 = (1.336\,9, 1.160\,7, 1.786\,6)^{\mathrm{T}}, \\ \max(\boldsymbol{v}_9) = 1.786\,6, \\ \boldsymbol{u}_9 = \boldsymbol{v}_9/\max(\boldsymbol{v}_9) = (0.748\,3, 0.649\,7, 1)^{\mathrm{T}}; \end{cases}$$

$$\begin{cases} \boldsymbol{v}_{10} = \boldsymbol{B}\boldsymbol{u}_9 = (1.336\,9, 1.160\,7, 1.786\,6)^{\mathrm{T}}, \\ \max(\boldsymbol{v}_{10}) = 1.786\,6, \\ \boldsymbol{u}_{10} = \boldsymbol{v}_{10}/\max(\boldsymbol{v}_{10}) = (0.748\,3, 0.649\,7, 1)^{\mathrm{T}}. \end{cases}$$

因为 $\|u_{10}-u_9\|_\infty=\max\{0,0,0\}=0$,

所以 $x_1=u_{10}=(0.748\,3,0.649\,7,1)^T$.

故 $\lambda_1-0.75=1.786\,6$,即

$$\lambda_1=1.786\,6+0.75=2.536\,6.$$

此题用原点平移法经10次迭代的结果与例1直接对 A 用规范化幂法经20次迭代的结果基本相同.

5.2.4 反幂法

5.2.4.1 计算最小模特征值及相应的特征向量

设矩阵 $A=(a_{ij})_{n\times n}$,$\det(A)\neq 0$,A 与 A^{-1}的特征值分别为

$$\lambda_i(i=1,2,\cdots,n),\quad \frac{1}{\lambda_i}(i=1,2,\cdots,n).$$

若 A 的特征值分布为

$$|\lambda_1|\geqslant|\lambda_2|\geqslant\cdots\geqslant|\lambda_{n-1}|>|\lambda_n|,$$

相应的特征向量为

$$x_1,x_2,\cdots,x_{n-1},x_n,$$

则 A^{-1}的特征值分布应为

$$\left|\frac{1}{\lambda_n}\right|>\left|\frac{1}{\lambda_{n-1}}\right|\geqslant\cdots\geqslant\left|\frac{1}{\lambda_2}\right|\geqslant\left|\frac{1}{\lambda_1}\right|,$$

相应的特征向量为

$$x_n,x_{n-1},\cdots,x_2,x_1.$$

定理 5.4 设矩阵 $A=(a_{ij})_{n\times n}$,$\det(A)\neq 0$,且

(1)A 有完全的特征向量系;

(2)A 的特征值分布为

$$|\lambda_1|\geqslant|\lambda_2|\geqslant\cdots\geqslant|\lambda_{n-1}|>|\lambda_n|>0.$$

对任给的 $u_0=v_0\neq 0$,对 A^{-1}作规范化幂法

$$\begin{cases}v_k=A^{-1}u_{k-1}, & \text{即 } Av_k=u_{k-1},\\ u_k=v_k/\max(v_k), & k=1,2,\cdots\end{cases}\tag{5-22}$$

则有

$$\lim_{k\to\infty}u_k=x_n/\max(x_n);\tag{5-23}$$

$$\lim_{k\to\infty}\max(v_k)=\frac{1}{\lambda_n}.\tag{5-24}$$

上述结论表明,规范化向量 u_k 的极限向量即是矩阵 A^{-1}的主特征向量,也是矩阵 A 的最小模特征值 λ_n 对应的特征向量;而迭代向量的最大模分量的极限则是 A^{-1}的主特征值,也是 A 的最小模特征值.

5.2.4.2 计算已知特征值所对应的特征向量

设 $A=(a_{ij})_{n\times n}$,已知 A 的某一特征值 λ_m 的近似值为 q,即 $\lambda_m\approx q$.一般说来,总会有

$$|\lambda_m-q|=\min_{1\leqslant i\leqslant n}|\lambda_i-\lambda_m|.$$

按原点平移法的思想，取 $p=q$，作 $B=A-qI$，则 B 的最小模特征值应为 λ_m-q，而对应的特征向量则与 A 的 λ_m 所对应的特征向量相同.

定理 5.5 设矩阵 $A=(a_{ij})_{n\times n}$ 的特征值系统为 $\lambda_i\ (i=1,2,\cdots,n)$，$x_i\ (i=1,2,\cdots,n)$，且

(1) $x_i\ (i=1,2,\cdots,n)$ 线性无关；

(2) $\lambda_m\approx q$，$(A-qI)^{-1}$ 存在，且满足

$$\frac{1}{|\lambda_m-q|}>\frac{1}{|\lambda_i-q|},\quad i=1,2,\cdots,n,\quad i\neq m;$$

(3) $u_0=\sum\limits_{i=1}^{n}a_i x_i\neq 0\ (a_m\neq 0)$.

对 $(A-qI)^{-1}$ 作规范化幂法：

$$\begin{cases}u_0=v\neq 0,\\ v_k=(A-qI)^{-1}u_{k-1},\quad 即 (A-qI)v_k=u_{k-1},\\ u_k=v_k/\max(v_k).\end{cases}\tag{5-26}$$

则有

$$\lim_{k\to\infty}u_k=x_m/\max(x_m);\tag{5-27}$$

$$\lim_{k\to\infty}\max(v_k)=\frac{1}{\lambda_m-q}.\tag{5-28}$$

即是说，规范化向量 u_k 的极限向量即是 λ_m 所对应的特征向量；而迭代向量的最大模分量的极限是 $\frac{1}{\lambda_m-q}$，从该数的倒数减去 q 即是 λ_m.

证 因为

$$\begin{aligned}v_k&=(A-qI)^{-1}u_{k-1}=(A-qI)^{-1}\frac{v_{k-1}}{\max(v_{k-1})}\\&=(A-qI)^{-1}\frac{(A-qI)^{-(k-1)}v_0}{\max[(A-qI)^{-(k-1)}v_0]}\\&=\frac{(A-qI)^{-k}v_0}{\max[(A-qI)^{-(k-1)}v_0]}\\&=\frac{(A-qI)^{-k}u_0}{\max[(A-qI)^{-(k-1)}u_0]},\end{aligned}$$

所以

$$\begin{aligned}u_k&=\frac{v_k}{\max(v_k)}=\frac{(A-qI)^{-k}u_0}{\max[(A-qI)^{-(k-1)}u_0]}\cdot\frac{\max[(A-qI)^{-(k+1)}u_0]}{\max[(A-qI)^{-k}u_0]}\\&=\frac{(A-qI)^{-k}u_0}{\max[(A-qI)^{-k}u_0]}.\end{aligned}$$

又因为

$$(A-qI)^k x_i=(\lambda_i-q)^k x_i,\quad i=1,2,\cdots,n,$$
$$(A-qI)^{-k}x_i=(\lambda_i-q)^{-k}x_i,\quad i=1,2,\cdots,n,$$

所以

$$\begin{aligned}(A-qI)^{-k}u_0&=(A-qI)^{-k}\Big(\sum_{i=1}^{n}a_i x_i\Big)\\&=\sum_{i=1}^{n}a_i(\lambda_i-q)^{-k}x_i\end{aligned}$$

$$=(\lambda_m-q)^{-k}\left[a_1\left(\frac{\lambda_m-q}{\lambda_1-q}\right)^k x_1+a_2\left(\frac{\lambda_m-q}{\lambda_2-q}\right)^k x_2\right.$$

$$\left.+\cdots+a_m x_m+\cdots+a_n\left(\frac{\lambda_m-q}{\lambda_n-q}\right)^k x_n\right].$$

$$\lim_{k\to\infty} u_k=\lim_{k\to\infty}\frac{a_1\left(\frac{\lambda_m-q}{\lambda_1-q}\right)^k x_1+\cdots+a_m x_m+\cdots+a_n\left(\frac{\lambda_m-q}{\lambda_n-q}\right)^k x_n}{\max\left[a_1\left(\frac{\lambda_m-q}{\lambda_1-q}\right)^k x_1+\cdots+a_m x_m+\cdots+a_n\left(\frac{\lambda_m-q}{\lambda_n-q}\right)^k x_n\right]}$$

$$=\frac{a_m x_m}{\max(a_m x_m)}=\frac{x_m}{\max(x_m)}.$$

同理可得

$$\lim_{k\to\infty}\max(v_k)=\frac{1}{\lambda_m-q}.$$

在反幂法迭代公式(5－26)中，求 v_k 可通过解方程组

$$(A-qI)v_k=u_{k-1} \tag{5－29}$$

求得. 但每迭代一次需要解一次方程组(5－29)，为了节省工作量，可事先用列主元消元法将 $A-qI$ 分解为两个三角矩阵的乘积，即用列主元三角分解法

$$P(A-qI)=LU.$$

其中，P 为某个置换阵. 于是，每次迭代求 v_k 只需解两个三角方程组

$$\begin{cases}Ly_k=Pu_{k-1},\\ Uv_k=y_k.\end{cases}$$

由此，反幂法迭代公式可写为

$$\begin{cases}Ly_k=Pu_{k-1},\\ u_k=y_k,\\ u_k=v_k/\max(v_k), \quad k=1,2,\cdots\\ u_0=v_0\neq 0 \quad (a_m\neq 0).\end{cases} \tag{5－30}$$

同理，用反幂法迭代公式(5－22)求 v_k，也可通过 LU 分解算法求解方程组 $Av_k=u_{k-1}$.

5.2.5 幂法与反幂法小结

(1)对 A 作规范化幂法.

设 $A=(a_{ij})_{n\times n}$ 的特征值分布为

$$|\lambda_1|>|\lambda_2|\geqslant\cdots\geqslant|\lambda_{n-1}|\geqslant|\lambda_n|$$

或 $$\lambda_1=\lambda_2=\cdots=\lambda_r, \quad |\lambda_r|>|\lambda_{r+1}|\geqslant\cdots\geqslant|\lambda_{n-1}|\geqslant|\lambda_n|,$$

则对 A 作规范化幂法：

$$\begin{cases}u_0=v_0\neq 0 \qquad (a_1\neq 0),\\ v_k=Au_{k-1},\\ u_k=v_k/\max(v_k), k=1,2,\cdots\end{cases}$$

从而可得

$$\begin{cases}\lim\limits_{k\to\infty}u_k=x_1/\max(x_1),\\ \lim\limits_{k\to\infty}\max(v_k)=\lambda_1.\end{cases}$$

(2)对 $A-pI$ 作规范化幂法(原点平移)：

$$\begin{cases}u_0=v_0\neq 0,\\ v_k=(A-pI)u_{k-1},\\ u_k=v_k/\max(v_k), \quad k=1,2,\cdots\end{cases}$$

若 $p=(\lambda_1+\lambda_n)/2$,则

$$\begin{cases}\lim\limits_{k\to\infty}u_k=x_1/\max(x_1),\\ \lim\limits_{k\to\infty}\max(v_k)=\lambda_1-p,\\ (\lambda_1-p)+p=\lambda_1;\end{cases}$$

若 $p=(\lambda_1+\lambda_{n-1})/2$,则

$$\begin{cases}\lim\limits_{k\to\infty}u_k=x_n/\max(x_n),\\ \lim\limits_{k\to\infty}\max(v_k)=\lambda_n-p,\\ (\lambda_n-p)+p=\lambda_n.\end{cases}$$

(3)对 A^{-1} 作规范化幂法.

设 A 与 A^{-1} 的特征值分布分别为

$$|\lambda_1|\geqslant|\lambda_2|\geqslant\cdots\geqslant|\lambda_{n-1}|>|\lambda_n|,$$

对应的特征向量为 $x_1,x_2,\cdots,x_n$;

$$\left|\frac{1}{\lambda_n}\right|>\left|\frac{1}{\lambda_{n-1}}\right|\geqslant\cdots\geqslant\left|\frac{1}{\lambda_2}\right|\geqslant\left|\frac{1}{\lambda_1}\right|,$$

对应的特征向量为 $x_n,x_{n-1},\cdots,x_1$. 对 A^{-1} 作规范化幂法：

$$\begin{cases}u_0=v_0\neq 0 \qquad (a_n\neq 0),\\ v_k=A^{-1}u_{k-1},\\ u_k=v_k/\max(v_k), \quad k=1,2,\cdots\end{cases}$$

$$\begin{cases}\lim\limits_{k\to\infty}u_k=x_n/\max(x_n),\\ \lim\limits_{k\to\infty}\max(v_k)=\dfrac{1}{\lambda_n}.\end{cases}$$

(4)对 $(A-qI)^{-1}$ 作规范化幂法($q\approx\lambda_m$)：

$$\begin{cases}u_0=v_0\neq 0 \qquad (a_m\neq 0),\\ v_k=(A-qI)^{-1}u_{k-1},\\ u_k=v_k/\max(v_k), \quad k=1,2,\cdots\end{cases}$$

$$\begin{cases}\lim\limits_{k\to\infty}u_k=x_m/\max(x_m),\\ \lim\limits_{k\to\infty}\max(v_k)=\dfrac{1}{\lambda_m-q}.\end{cases}$$

习题 5

*1. 用规范化幂法计算下列矩阵的主特征值及对应的特征向量. 当特征值为 3 位小

数稳定时终止迭代.

$$(1)\boldsymbol{A}_1=\begin{bmatrix}7 & 3 & 2\\ 3 & 4 & -1\\ -2 & -1 & 3\end{bmatrix},\qquad (2)\boldsymbol{A}_2=\begin{bmatrix}3 & -4 & 3\\ -4 & 6 & 3\\ 3 & 3 & 1\end{bmatrix}.$$

*2. 用原点平移法计算矩阵 $\boldsymbol{A}$ 的主特征值,其中

$$\boldsymbol{A}=\begin{bmatrix}-3 & 1 & 0\\ 1 & -3 & -3\\ 0 & -3 & 4\end{bmatrix}.$$

取 $p=-4$,当特征值有 3 位小数稳定时终止迭代.

*3. 设 $\boldsymbol{A}=\begin{bmatrix}2 & 1 & 0\\ 1 & 3 & 1\\ 0 & 1 & 4\end{bmatrix}$,用反幂法求 $\boldsymbol{A}$ 的最接近于 $q=1.2679$ 的特征向量,迭代次数为 2 次.

注　打 * 号的习题在本书附录Ⅱ作了详解.

第 6 章　插值法与曲线拟合

在许多实际问题中，为了研究某些变量间的函数关系，往往是通过实测数据来选择一个比较合理且易计算的近似函数代替原函数的. 有各种不同的求取近似函数的方法. 在本章中，我们介绍两个在科学研究和工程计算中最常用的方法——多项式插值与曲线拟合.

6.1　一元代数函数插值

6.1.1　插值问题

定义 6.1　给定实测数据

$$(x_i, y_i)_{i=0,1,\cdots,n},$$

其中 $x_i \neq x_j (i \neq j)$，且 $y_i = f(x_i)$；

给定函数系

$$\{\varphi_k(x)\}_{k=0,1,\cdots,n},$$

其中 $\varphi_k(x)$ 均为次数不超过 n 的多项式，且 $\{\varphi_k(x)\}$ 在点集 $\{x_0, x_1, \cdots, x_n\}$ 上线性无关，即

$$\begin{vmatrix} \varphi_0(x_0) & \varphi_1(x_0) & \cdots & \varphi_n(x_0) \\ \varphi_0(x_1) & \varphi_1(x_1) & \cdots & \varphi_n(x_1) \\ \vdots & \vdots & & \vdots \\ \varphi_0(x_n) & \varphi_1(x_n) & \cdots & \varphi_n(x_n) \end{vmatrix} \neq 0. \tag{6-1}$$

求 $p_n(x) = \sum_{k=0}^{n} c_k \varphi_k(x)$，使得

$$p_n(x_i) = y_i = f(x_i), \quad i = 0, 1, \cdots, n. \tag{6-2}$$

我们称上述问题为一元代数插值问题，并称 $x_i (i=0,1,\cdots,n)$ 为插值结点；$\varphi_k(x)(k=0,1,\cdots,n)$ 为插值基函数；$p_n(x)$ 为插值多项式；$f(x)$ 为被插值函数；式(6-2)为插值条件；$[a,b]$为插值区间，其中 $a=\min\limits_{0\leqslant i\leqslant n} x_i$，$b=\max\limits_{0\leqslant i\leqslant n} x_i$.

通常一个插值问题应包含三个方面的内容：

(1)构造插值多项式 $p_n(x)$，且满足插值条件(6-2)；

(2)对给定的插值点 $x \in [a,b]$，$x \neq x_i$，计算 $f(x) \approx p_n(x)$；

(3)估计余项 $R(x) = f(x) - p_n(x)$.

6.1.2　插值多项式的存在惟一性

由插值条件 $p_n(x_i) = y_i (i=0,1,\cdots,n)$ 得到线性方程组

$$\sum_{k=0}^{n} c_k \varphi_k(x_i) = y_i, \quad i = 0, 1, \cdots, n.$$

展开上式得

$$
\begin{gathered}
c_0\varphi_0(x_0)+c_1\varphi_1(x_0)+\cdots+c_n\varphi_n(x_0)=y_0,\\
c_0\varphi_0(x_1)+c_1\varphi_1(x_1)+\cdots+c_n\varphi_n(x_1)=y_1,\\
\vdots\\
c_0\varphi_0(x_n)+c_1\varphi_1(x_n)+\cdots+c_n\varphi_n(x_n)=y_n.
\end{gathered}
$$

即

$$
\begin{bmatrix}
\varphi_0(x_0) & \varphi_1(x_0) & \cdots & \varphi_n(x_0)\\
\varphi_0(x_1) & \varphi_1(x_1) & \cdots & \varphi_n(x_1)\\
\vdots & \vdots & & \vdots\\
\varphi_0(x_n) & \varphi_1(x_n) & \cdots & \varphi_n(x_n)
\end{bmatrix}
\begin{bmatrix} c_0\\ c_1\\ \vdots\\ c_n \end{bmatrix}
=
\begin{bmatrix} y_0\\ y_1\\ \vdots\\ y_n \end{bmatrix}.
$$

据定义 6.1，$\{\varphi_k(x)\}$在点集$\{x_0,x_1,\cdots,x_n\}$上线性无关，即以上方程组系数矩阵的行列式不为零，所以该方程组存在惟一解 $c_k(k=0,1,\cdots,n)$，故 $p_n(x)=\sum\limits_{k=0}^{n}c_k\varphi_k(x)$存在且惟一.

6.2 拉格朗日(Lagrange)插值方法

6.2.1 插值基函数

定义 6.2 令定义 6.1 中的基函数 $\varphi_k(x)=l_k(x)(k=0,1,\cdots,n)$，且

$$
l_k(x_i)=\begin{cases}1, & i=k,\\ 0, & i\neq k,\end{cases}\quad i=0,1,\cdots,n. \tag{6-3}
$$

称 $l_k(x)(k=0,1,\cdots,n)$为 Lagrange 插值基函数.

容易证明基函数 $l_k(x)$具有以下性质：

(1)基函数 $l_k(x)$的表达式为

$$
l_k(x)=\frac{(x-x_0)\cdots(x-x_{k-1})(x-x_{k+1})\cdots(x-x_n)}{(x_k-x_0)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-x_n)}.
$$

证 据式(6-3)$l_k(x_i)=0,i\neq k$，知 $l_k(x)$具有 n 个零点 $x_i(i=0,1,\cdots,k-1,k+1,\cdots,n)$，所以

$$
l_k(x)=A(x-x_0)\cdots(x-x_{k-1})(x-x_{k+1})\cdots(x-x_n).
$$

又由 $l_k(x_k)=1$，得

$$
A(x_k-x_0)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-x_n)=1,
$$

$$
A=\frac{1}{(x_k-x_0)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-x_n)},
$$

因而

$$
l_k(x)=\frac{(x-x_0)\cdots(x-x_{k-1})(x-x_{k+1})\cdots(x-x_n)}{(x_k-x_0)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-x_n)}.
$$

(2)基函数序列$\{l_k(x)\}_{k=0,1,\cdots,n}$在点集$\{x_0,x_1,\cdots,x_n\}$上线性无关.

证 据式(6-3)有

$$\begin{vmatrix} l_0(x_0) & l_1(x_0) & \cdots & l_n(x_0) \\ l_0(x_1) & l_1(x_1) & \cdots & l_n(x_1) \\ \vdots & \vdots & & \vdots \\ l_0(x_n) & l_1(x_n) & \cdots & l_n(x_n) \end{vmatrix} = \begin{vmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{vmatrix} = 1 \neq 0,$$

所以$\{l_k(x)\}_{k=0,1,\cdots,n}$在点集$\{x_0,x_1,\cdots,x_n\}$上线性无关.

6.2.2 Lagrange 插值多项式

定义 6.3 形如

$$L_n(x)=\sum_{k=0}^{n} y_k l_k(x)$$

的 n 次插值多项式称为 Lagrange 插值多项式,其中 $y_k(k=0,1,\cdots,n)$为实测数据的函数值,$l_k(x)(k=0,1,\cdots,n)$为 Lagrange 插值基函数.

易知 $L_n(x)$具有以下性质:

(1)$L_n(x)$满足插值条件 $L_n(x_i)=y_i, i=0,1,\cdots,n$.

事实上,据式(6-3)有

$$L_n(x_i)=\sum_{k=0}^{n} y_k l_k(x_i)=y_i, \quad i=0,1,\cdots,n.$$

(2)$L_n(x)$惟一存在.

由于 $L_n(x)$满足插值条件,且基函数序列$\{l_k(x)\}$在点集$\{x_0,x_1,\cdots,x_n\}$上线性无关,故据 6.1.2 的结论,$L_n(x)$惟一存在.

我们还可以用另外的方法证明 $L_n(x)$的存在惟一性.

设 n 次多项式 $\psi_n(x)\neq L_n(x)$,且 $\psi_n(x)$满足插值条件,即 $\psi_n(x_i)=y_i, i=0,1,\cdots,n$.

令
$$F_n(x)=L_n(x)-\psi_n(x), \tag{6-4}$$
且
$$F_n(x_i)=L_n(x_i)-\psi_n(x_i)=0, \quad i=0,1,\cdots,n. \tag{6-5}$$

由式(6-4)知,$F_n(x)$是次数不超过 n 次的多项式;又由式(6-5)知,$F_n(x)$有 $n+1$ 个零点,这显然不符合代数基本定理,所以 $F_n(x)\equiv 0$,即

$$\psi_n(x)=L_n(x).$$

6.2.3 $L_n(x)$的两种表达式

(1)紧凑形式:

$$l_k(x)=\frac{(x-x_0)\cdots(x-x_{k-1})(x-x_{k+1})\cdots(x-x_n)}{(x_k-x_0)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-x_n)}$$

$$=\prod_{\substack{j=0\\ j\neq k}}^{n}\frac{x-x_j}{x_k-x_j},$$

$$L_n(x)=\sum_{k=0}^{n} y_k l_k(x)=\sum_{k=0}^{n}\Big(\prod_{\substack{j=0\\ j\neq k}}^{n}\frac{x-x_j}{x_k-x_j}\Big)y_k. \tag{6-6}$$

称式(6-6)为 $L_n(x)$的紧凑形式.

(2)带导数的形式：

设

$$\omega_{n+1}(x)=(x-x_0)(x-x_1)\cdots(x-x_n)=\prod_{j=0}^{n}(x-x_j),$$

则

$$\omega'_{n+1}(x)=\prod_{j=1}^{n}(x-x_j)+\prod_{\substack{j=0\\ j\neq 1}}^{n}(x-x_j)+\cdots+\prod_{\substack{j=0\\ j\neq k}}^{n}(x-x_j)+\cdots+\prod_{j=0}^{n-1}(x-x_j),$$

$$\omega'_{n+1}(x_k)=\prod_{\substack{j=0\\ j\neq k}}^{n}(x_k-x_j),$$

所以

$$l_k(x)=\frac{\omega_{n+1}(x)}{(x-x_k)\omega'_{n+1}(x_k)},$$

$$L_n(x)=\sum_{k=0}^{n}y_k\frac{\omega_{n+1}(x)}{(x-x_k)\omega'_{n+1}(x_k)}. \tag{6-7}$$

称式(6－7)为 $L_n(x)$ 的带导数的形式.

6.2.4 插值应用举例

对于 Lagrange 插值多项式，通常用到 $n=1,2$ 的情形.

当 $n=1$ 时，实测数据为 $(x_i,y_i)_{i=0,1}$，则 Lagrange 插值多项式为

$$L_1(x)=y_0l_0(x)+y_1l_1(x)=\frac{x-x_1}{x_0-x_1}y_0+\frac{x-x_0}{x_1-x_0}y_1.$$

令 $L_1(x)=y$，则上式即是通过两点的直线方程.

下面列出直线方程的几种形式：

(1)点斜式：$y-y_0=\frac{y_1-y_0}{x_1-x_0}(x-x_0)$；

(2)两点式：$\frac{y-y_0}{y_1-y_0}=\frac{x-x_0}{x_1-x_0}$；

(3)对称式：$y=\frac{x-x_1}{x_0-x_1}y_0+\frac{x-x_0}{x_1-x_0}y_1$.

由此可见，Lagrange 一次插值多项式所形成的直线方程即是通过两点的直线对称式方程.

当 $n=2$ 时，实测数据为 $(x_i,y_i)_{i=0,1,2}$，则 Lagrange 插值多项式为

$$L_2(x)=y_0l_0(x)+y_1l_1(x)+y_2l_2(x).$$

其中

$$l_0(x)=\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)},$$

$$l_1(x)=\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)},$$

$$l_2(x)=\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}.$$

例1 已知$\sqrt{4}=2$，$\sqrt{9}=3$，$\sqrt{16}=4$，求$\sqrt{7}$的近似值.

解 令$y=\sqrt{x}$，列表6－1.

表6－1

x	$x_0=4$	$x_1=9$	$x_2=16$
$y=\sqrt{x}$	$y_0=2$	$y_1=3$	$y_2=4$

(1)用线性插值多项式.

在表6－1所列的三组数据中，可以任取两组数据构造线性插值多项式$L_1(x)$. 鉴于插值点$x=7$所处的位置，宜于选取(x_0,y_0)，(x_1,y_1)构造$L_1(x)$.

$$\begin{aligned}L_1(x)&=l_0(x)y_0+l_1(x)y_1\\&=\frac{x-9}{4-9}\times 2+\frac{x-4}{9-4}\times 3\\&=-\frac{2}{5}(x-9)+\frac{3}{5}(x-4).\end{aligned}$$

所以
$$\sqrt{7}\approx L_1(7)=2.6.$$

(2)用全部数据$(x_i,y_i)_{i=0,1,2}$构造二次插值多项式$L_2(x)$.

$$\begin{aligned}L_2(x)&=l_0(x)y_0+l_1(x)y_1+l_2(x)y_2\\&=\frac{(x-9)(x-16)}{(4-9)(4-16)}\times 2+\frac{(x-4)(x-16)}{(9-4)(9-16)}\times 3\\&\quad+\frac{(x-4)(x-9)}{(16-4)(16-9)}\times 4,\end{aligned}$$

所以
$$\sqrt{7}\approx L_2(7)=\frac{3}{5}+\frac{81}{35}-\frac{2}{7}=2.628\,6.$$

例2 已知正弦函数$\sin x$的部分数据如表6－2所示，求$\sin 40^\circ$的近似值.

表6－2

x	$x_0=30^\circ$	$x_1=45^\circ$	$x_2=60^\circ$
$y=\sin x$	$y_0=0.5$	$y_1=0.707\,1$	$y_2=0.866\,0$

解 (1)取(x_0,y_0)，(x_1,y_1)构造$L_1(x)$.

因为
$$x_0=30^\circ=\frac{\pi}{6},\quad x_1=45^\circ=\frac{\pi}{4},\quad x=40^\circ=\frac{2}{9}\pi,$$

所以
$$\begin{aligned}L_1(x)&=l_0(x)y_0+l_1(x)y_1\\&=\frac{x-\frac{\pi}{4}}{\frac{\pi}{6}-\frac{\pi}{4}}\times 0.5+\frac{x-\frac{\pi}{6}}{\frac{\pi}{4}-\frac{\pi}{6}}\times 0.707\,1,\end{aligned}$$

$$\sin 40^\circ=\sin\left(\frac{2\pi}{9}\right)\approx L_1\left(\frac{2\pi}{9}\right)=0.638\,0.$$

(2)用全部数据(x_i,y_i)，$i=0,1,2$构造$L_2(x)$.

$$L_2(x)=l_0(x)y_0+l_1(x)y_1+l_2(x)y_2,$$

其中
$$l_0(x)=\frac{(x-\frac{\pi}{4})(x-\frac{\pi}{3})}{(\frac{\pi}{6}-\frac{\pi}{4})(\frac{\pi}{6}-\frac{\pi}{3})},$$

$$l_1(x)=\frac{(x-\frac{\pi}{6})(x-\frac{\pi}{3})}{(\frac{\pi}{4}-\frac{\pi}{6})(\frac{\pi}{6}-\frac{\pi}{3})},$$

$$l_2(x)=\frac{(x-\frac{\pi}{6})(x-\frac{\pi}{4})}{(\frac{\pi}{3}-\frac{\pi}{6})(\frac{\pi}{3}-\frac{\pi}{4})}.$$

因为
$$l_0(\frac{2\pi}{9})=\frac{2}{9},\quad l_1(\frac{2\pi}{9})=\frac{8}{9},\quad l_2(\frac{2\pi}{9})=-\frac{1}{9},$$

所以
$$L_2(\frac{2\pi}{9})=\frac{2}{9}\times 0.5+\frac{8}{9}\times 0.7071-\frac{1}{9}\times 0.8660=0.6434,$$

$$\sin 40^\circ=\sin(\frac{2\pi}{9})\approx L_2(\frac{2\pi}{9})=0.6434.$$

6.2.5 插值余项

若在$[a,b]$上用$L_n(x)$近似$f(x)$，则其截断误差$R_n(x)=f(x)-L_n(x)$称为插值多项式$L_n(x)$的余项.

显然
$$R_n(x_i)=f(x_i)-L_n(x_i)=0,\quad i=0,1,\cdots,n,$$
而对$\forall x\neq x_i(i=0,1,\cdots,n)$有
$$R_n(x)\neq 0.$$

估计$|R_n(x)|$的值可得到$f(x)$的近似值$L_n(x)$的误差估计. 关于余项$R_n(x)$的估计，有以下定理成立.

定理 6.1 设

(1)$f(x)$在$[a,b]$上连续，$f^{(n+1)}(x)$在区间(a,b)内存在；

(2)节点x_i满足
$$a\leqslant x_0\leqslant x_1\leqslant\cdots\leqslant x_n\leqslant b;$$

(3)$L_n(x)$为$f(x)$的 Lagrange 插值多项式.

由此可得
$$R_n(x)=f(x)-L_n(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!}\omega_{n+1}(x),\quad \xi\in(a,b).$$

证 因为
$$R_n(x_i)=0,\quad i=0,1,\cdots,n,$$
所以
$$R_n(x)=K(x)(x-x_0)\cdots(x-x_n)$$
$$=K(x)\omega_{n+1}(x).$$

设
$$\varphi(t)=f(t)-L_n(t)-K(x)(t-x_0)\cdots(t-x_n),$$
则$\varphi(t)=0$，$t=x_0,\cdots,x_n,x$，即$\varphi(t)$在区间$[a,b]$内至少有$n+2$个零点. 因此：

$\varphi'(t)$在区间$[a,b]$内至少有$n+1$个零点；

$\varphi''(t)$在区间$[a,b]$内至少有 n 个零点;

$$\vdots$$

$\varphi^{(n+1)}(t)$在区间$[a,b]$内至少有一个零点. 记该零点为ξ,于是有

$$\varphi^{(n+1)}(\xi)=f^{(n+1)}(\xi)-K(x)(n+1)!\ =0.$$

所以

$$K(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!},$$

$$R_n(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!}\omega_{n+1}(x).$$

在估计 $R_n(x)$时遇到的困难是,由于$\xi\in(a,b)$不能具体给出,故 $f^{(n+1)}(\xi)$无法计算,通常是估计出

$$\max_{a\leqslant x\leqslant b}|f^{(n+1)}(x)|=M_{n+1},$$

于是有

$$|R_n(x)|\leqslant\frac{M_{n+1}}{(n+1)!}|\omega_{n+1}(x)|.$$

在例 2 中,因为

$$\sin 40^\circ=L_2\left(\frac{2\pi}{9}\right)=0.643\,4,$$

则

$$R_2(40^\circ)=R_2\left(\frac{2\pi}{9}\right)$$

$$\leqslant\frac{M_3}{3!}\left(\frac{2\pi}{9}-\frac{\pi}{6}\right)\left(\frac{2\pi}{9}-\frac{\pi}{4}\right)\left(\frac{2\pi}{9}-\frac{\pi}{3}\right).$$

而

$$M_3=\max_{x\in(a,b)}|f'''(x)|$$

$$=\max|\cos x|\leqslant 1,$$

所以

$$R_2(40^\circ)\leqslant\frac{1}{6}\times 0.174\,5\times 0.087\,3\times 0.349\,1$$

$$=0.000\,866$$

$$=0.866\times 10^{-3}<0.5\times 10^{-2}.$$

故

$$\sin 40^\circ\approx 0.64(\text{准确至两位小数}).$$

6.2.6 Lagrange 插值方法评价

Lagrange 插值方法是最基本的插值方法,其插值公式形式对称,便于记忆,但在实际应用中有诸多不便.

首先,它不是一种自适应的算法,即是说 Lagrange 插值多项式无法实现逐步递推.

例如,已知$(x_i,y_i)_{i=0,1}$,则

$$L_1(x)=l_0(x)y_0+l_1(x)y_1$$

$$=\frac{x-x_1}{x_0-x_1}y_0+\frac{x-x_0}{x_1-x_0}y_1;$$

若增加一个节点$(x_i,y_i)_{i=0,1,2}$,则

$$L_2(x)=l_0(x)y_0+l_1(x)y_1+l_2(x)y_2$$

$$=\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}y_0+\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}y_1+\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}y_2.$$

显然 $L_2(x)$不能由 $L_1(x)$递推得到.

其次,由插值余项 $R_n(x)$估计 $L_n(x)$的误差十分困难. 据定理 6.1, $|R_n(x)| \leqslant \frac{M_{n+1}}{(n+1)!}|\omega_{n+1}(x)|$,其中 $M_{n+1}=\max\limits_{\xi\in(a,b)}|f^{(n+1)}(\xi)|$. 当 $f(x)$无表达式时,采用此方法根本无法估计 M_{n+1}. 一般说来,即使给出 $f(x)$的表达式,估计 M_{n+1}也是十分困难的.

下面我们将介绍一种新的插值方法——Newton 均差插值法. 它是一种自适应算法,具有便于估计误差等优点.

6.3 Newton 均差插值方法

6.3.1 均差与均差表

定义 6.4 设$(x_i,y_i)_{i=0,1,\cdots,n}$,其中 $x_i\neq x_j(i\neq j)$,则称

$$f[x_{i-1},x_i]=\frac{f(x_i)-f(x_{i-1})}{x_i-x_{i-1}},\quad i=1,2,\cdots,n$$

为 $f(x)$关于结点 x_{i-1},x_i 的一阶均差;称

$$f[x_{i-2},x_{i-1},x_i]=\frac{f[x_{i-1},x_i]-f[x_{i-2},x_{i-1}]}{x_i-x_{i-2}},\quad i=2,3,\cdots,n$$

为 $f(x)$关于结点 x_{i-2},x_{i-1},x_i 的二阶均差;称

$$f[x_{i-k},x_{i-k+1},\cdots,x_i]=\frac{f[x_{i-k+1},\cdots,x_i]-f[x_{i-k},\cdots,x_{i-1}]}{x_i-x_{i-k}},\quad i=k,k+1,\cdots,n$$

为 $f(x)$关于结点 $x_{i-k},x_{i-k+1},\cdots,x_i$ 的 k 阶均差.

令 $k=1,2,\cdots,n$,我们便得到由一阶均差,二阶均差……直至 n 阶均差构成的均差表(见表 6-3).

表 6-3

x_i	$f(x)$	$f[x_{i-1},x_i]$	$f[x_{i-2},x_{i-1},x_i]$		$f[x_{i-k},x_{i-k+1},\cdots,x_i]$		
x_0	$f(x_0)$						
x_1	$f(x_1)$	$f[x_0,x_1]$					
x_2	$f(x_2)$	$f[x_1,x_2]$	$f[x_0,x_1,x_2]$				
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$			
x_k	$f(x_k)$	$f[x_{k-1},x_k]$	$f[x_{k-2},x_{k-1},x_k]$	$\cdots$	$f[x_0,x_1,\cdots,x_k]$		
$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\ddots$	
x_n	$f(x_n)$	$f[x_{n-1},x_n]$	$f[x_{n-2},x_{n-1},x_n]$	$\cdots$	$f[x_{n-k},x_{n-k+1},\cdots,x_n]$	$\cdots$	$f[x_0,x_1,\cdots,x_n]$

6.3.2 Newton 均差插值多项式

设$(x_i,y_i)_{i=0,1,\cdots,n}$,$x_i\neq x_j(i\neq j)$;$x$ 为插值区间$[a,b]$上的任一点,依次作一阶,二阶,$\cdots$,$n+1$ 阶均差,即

$$f[x,x_0]=\frac{f(x_0)-f(x)}{x_0-x},$$

$$f[x,x_0,x_1]=\frac{f[x_0,x_1]-f[x,x_0]}{x_1-x},$$

$$f[x,x_0,x_1,x_2]=\frac{f[x_0,x_1,x_2]-f[x,x_0,x_1]}{x_2-x},$$

$$\vdots$$

$$f[x,x_0,x_1,\cdots,x_n]=\frac{f[x_0,x_1,\cdots,x_n]-f[x,x_0,\cdots,x_{n-1}]}{x_n-x}.$$

从以上各式依次解得

$$f(x)=f(x_0)+f[x,x_0](x-x_0),$$

$$f[x,x_0]=f[x_0,x_1]+f[x,x_0,x_1](x-x_1),$$

$$f[x,x_0,x_1]=f[x_0,x_1,x_2]+f[x,x_0,x_1,x_2](x-x_2),$$

$$\vdots$$

$$f[x,x_0,\cdots,x_{n-1}]=f[x_0,x_1,\cdots,x_n]+f[x,x_0,x_1,\cdots,x_n](x-x_n).$$

依次将后式代入前式得

$$\begin{aligned}f(x)=&f(x_0)+f[x_0,x_1](x-x_0)+f[x_0,x_1,x_2](x-x_0)(x-x_1)\\&+\cdots+f[x_0,x_1,\cdots,x_n](x-x_0)(x-x_1)\cdots(x-x_{n-1})\\&+f[x,x_0,x_1,\cdots,x_n](x-x_0)(x-x_1)\cdots(x-x_n).\end{aligned}$$

记 $\omega_k(x)=(x-x_0)\cdots(x-x_{k-1})\quad(k=1,2,\cdots,n+1)$,

则

$$\begin{aligned}f(x)=&f(x_0)+f[x_0,x_1]\omega_1(x)+f[x_0,x_1,x_2]\omega_2(x)\\&+\cdots+f[x_0,x_1,\cdots,x_n]\omega_n(x)+f[x,x_0,x_1,\cdots,x_n]\omega_{n+1}(x).\end{aligned}$$

令

$$N_n(x)=\sum_{k=0}^{n}f[x_0,x_1,\cdots,x_k]\omega_k(x),\tag{6-8}$$

$$R_n(x)=f[x,x_0,x_1,\cdots,x_n]\omega_{n+1}(x),\tag{6-9}$$

则有

$$f(x)=N_n(x)+R_n(x).\tag{6-10}$$

我们称 n 次多项式 $N_n(x)$ 为 Newton 均差插值多项式,$R_n(x)$ 为其插值余项.

从表 6-3 看出,Newton 均差插值多项式 $N_n(x)$ 的各项系数即是均差表对角线上的各阶均差.

$N_n(x)$ 具有如下性质:

(1)

$$N_n(x)=\sum_{k=0}^{n}c_k\,\omega_k(x),$$

其中 $\omega_k(x)(k=0,1,\cdots,n)$ 是构成 $N_n(x)$ 的基函数,而组合常数 $c_k=f[x_0,x_1,\cdots,x_k](k=0,1,\cdots,n)$.

(2) $N_n(x)$ 满足插值条件,即

$$N_n(x_i)=y_i,\quad i=0,1,\cdots,n.$$

事实上,$R_n(x_i)=0,i=0,1,\cdots,n$,由式(6-10)知

$$N_n(x_i)=f(x_i)-R_n(x_i)=f(x_i)=y_i,\quad i=0,1,\cdots,n.$$

(3) $N_n(x)$ 惟一存在.

据性质(1),有 $N_n(x)=\sum_{k=0}^{n}c_k\omega_k(x)$. 以下只需证明基函数列 $\{\omega_k(x)\}_{k=0,1,\cdots,n}$ 在点集 $\{x_0,x_1,\cdots,x_n\}$ 上线性无关. 事实上是证 $\{\omega_k(x)\}_{k=0,1,\cdots,n}$ 在 $\{x_0,x_1,\cdots,x_n\}$ 上构成的行列式不为零,即

$$\begin{vmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & x_1-x_0 & 0 & \cdots & 0 \\ 1 & x_2-x_0 & (x_2-x_0)(x_2-x_1) & & \vdots \\ \vdots & \vdots & \vdots & \ddots & \\ 1 & x_n-x_0 & (x_n-x_0)(x_n-x_1) & \cdots & (x_n-x_0)(x_n-x_1)\cdots(x_n-x_{n-1}) \end{vmatrix}$$

$$=1\cdot(x_1-x_0)(x_2-x_0)(x_2-x_1)\cdots(x_n-x_0)(x_n-x_1)\cdots(x_n-x_{n-1})\neq 0$$

$$(x_i\neq x_j,\quad i\neq j).$$

由 6.1.2 的证明可知 $N_n(x)$ 惟一存在.

(4)插值余项 $R_n(x)$ 的估计式为

$$\begin{aligned} R_n(x)&=f[x,x_0,x_1,\cdots,x_n]\omega_{n+1}(x) \\ &=f[x_0,x_1,\cdots,x_n,x]\omega_{n+1}(x)^{注}. \end{aligned}$$

鉴于 $f[x_0,x_1,\cdots,x_n,x]$ 中 x 的不确定性,通常采用两种估计 $R_n(x)$ 的方案:

1° 增加一个节点 x_{n+1},以 x_{n+1} 代替 x,即

$$\begin{cases} N_n(x)=\sum_{k=0}^{n}f[x_0,x_1,\cdots,x_k]\omega_k(x), \\ R_n(x)\approx f[x_0,x_1,\cdots,x_n,x_{n+1}]\omega_{n+1}(x); \end{cases}$$

2° 插值公式退后一阶,以 x_n 代替 x,即

$$\begin{cases} N_{n-1}(x)=\sum_{k=0}^{n-1}f[x_0,x_1,\cdots,x_k]\omega_k(x), \\ R_{n-1}(x)\approx f[x_0,x_1,\cdots,x_{n-1},x_n]\omega_n(x). \end{cases}$$

注 等式 $f[x,x_0,\cdots,x_n]=f[x_0,\cdots,x_n,x]$ 的依据将在 6.3.3 中的性质 2 中给出.

例 3 表 6－4 为正弦函数 $\sin x$ 的部分数据,求 $\sin 40^\circ=\sin(\frac{2}{9}\pi)$ 的近似值($\frac{2}{9}\pi\approx 0.698\,1$).

表 6－4

x	$\sin x$	x	$\sin x$
$\frac{\pi}{6}$	0.5	$\frac{\pi}{3}$	0.866 0
$\frac{\pi}{4}$	0.707 1	$\frac{\pi}{2}$	1

解 先构造均差表如表 6－5 所示.

表 6－5

x_i	$\sin x_i$	一阶均差	二阶均差	三阶均差
0.523 6	0.5			
0.785 4	0.707 1	0.791 1		
1.047 2	0.866 0	0.607 0	−0.351 6	
1.570 8	1	0.255 9	−0.447 0	−0.091 1

以表 6－5 中对角线上的各阶均差为系数，建立各阶 Newton 均差插值多项式，并估计误差.

(1)
$$N_1(x)=f(x_0)+f[x_0,x_1](x-x_0)$$
$$=0.5+0.791\,1(x-0.523\,6),$$
$$N_1(0.698\,1)=0.5+0.791\,1(0.698\,1-0.523\,6)$$
$$=0.638\,0,$$
$$R_1(x)\approx f[x_0,x_1,x_2](x-x_0)(x-x_1)$$
$$=-0.351\,6(x-0.523\,6)(x-0.785\,4),$$
$$|R_1(0.698\,1)|=0.535\,6\times10^{-2}<0.5\times10^{-1},$$

所以 $\sin40^\circ\approx N_1(0.698\,1)=0.638\,0$（准确至第一位小数）.

(2)
$$N_2(x)=N_1(x)+f[x_0,x_1,x_2](x-x_0)(x-x_1)$$
$$=N_1(x)+R_1(x),$$
$$N_2(0.698\,1)=0.638\,0+0.535\,6\times10^{-2}$$
$$=0.643\,4,$$
$$R_2(x)\approx f[x_0,x_1,x_2,x_3](x-x_0)(x-x_1)(x-x_2),$$
$$|R_2(0.698\,1)|\approx0.091\,1\times0.174\,5\times0.087\,3\times0.349\,1$$
$$=0.484\,5\times10^{-3}<0.5\times10^{-3},$$

所以 $\sin40^\circ\approx N_2(0.698\,1)=0.643\,4$（准确至第三位小数）.

例 4 表 6－6 为 x_i，$f(x_i)$ 的部分数据，求 $f(4.01)$.

表 6－6

x_i	$f(x_i)$	x_i	$f(x_i)$
4.000 2	0.602 081 7	4.023 3	0.604 582 4
4.010 4	0.603 187 7	4.029 4	0.605 240 4

解 构造均差表如表 6－7 所示.

表 6－7

x_i	$f(x_i)$	一阶均差	二阶均差	三阶均差
4.000 2	0.602 081 7			
4.010 4	0.603 187 7	0.108 431 4		

续表 6－7

x_i	$f(x_i)$	一阶均差	二阶均差	三阶均差
4.023 3	0.604 582 4	0.108 116 3	$\underline{-0.013\,640\,7}$	
4.029 4	0.605 240 4	0.107 868 9	−0.013 021 1	$\underline{0.012\,119\,2}$

$$N_2(x)=0.602\,081\,7+0.108\,431\,4(x-4.000\,2)$$
$$-0.013\,640\,7\times(x-4.000\,2)(x-4.010\,1),$$
$$N_2(4.01)=0.603\,144\,4,$$
$$R_2(x)\approx 0.021\,219\,2(x-4.000\,2)(x-4.010\,4)(x-4.023\,3),$$
$$|R_2(4.01)|=0.1\times 10^{-8}<0.5\times 10^{-8},$$

所以 $f(4.01)\approx N_2(4.01)=0.\underline{603\,144\,4}$.

6.3.3 均差的性质

(1)
$$f[x_0,x_1,\cdots,x_k]=\sum_{j=0}^{k}\frac{f(x_j)}{\omega_{k+1}'(x_j)}, \tag{6-11}$$

其中 $\omega_{k+1}'(x_j)=(x_j-x_0)\cdots(x_j-x_{j-1})(x_j-x_{j+1})\cdots(x_j-x_k)$.

容易验证 $k=1,2$ 时,式(6－11)成立.

当 $k=1$ 时,有

$$f[x_0,x_1]=\frac{f(x_1)-f(x_0)}{x_1-x_0}=\frac{f(x_0)}{x_0-x_1}+\frac{f(x_1)}{x_1-x_0},$$

而
$$\omega_2(x)=(x-x_0)(x-x_1),$$
$$\omega_2'(x_0)=x_0-x_1,$$
$$\omega_2'(x_1)=x_1-x_0.$$

所以
$$f[x_0,x_1]=\frac{f(x_0)}{\omega_2'(x_0)}+\frac{f(x_1)}{\omega_2'(x_1)}.$$

当 $k=2$ 时,有

$$f[x_0,x_1,x_2]=\frac{f[x_1,x_2]-f[x_0,x_1]}{x_2-x_0}.$$

将
$$f[x_1,x_2]=\frac{f(x_2)-f(x_1)}{x_2-x_1},\quad f[x_0,x_1]=\frac{f(x_1)-f(x_0)}{x_1-x_0}$$

代入上式得

$$f[x_0,x_1,x_2]=\frac{f(x_0)}{(x_0-x_1)(x_0-x_2)}+\frac{f(x_1)}{(x_1-x_0)(x_1-x_2)}+\frac{f(x_2)}{(x_2-x_0)(x_2-x_1)}.$$

而
$$\omega_3(x_0)=(x-x_0)(x-x_1)(x-x_2),$$
$$\omega_3'(x_0)=(x_0-x_1)(x_0-x_2),$$
$$\omega_3'(x_1)=(x_1-x_0)(x_1-x_2),$$
$$\omega_3'(x_2)=(x_2-x_0)(x_2-x_1).$$

所以
$$f[x_0,x_1,x_2]=\frac{f(x_0)}{\omega_3'(x_0)}+\frac{f(x_1)}{\omega_3'(x_1)}+\frac{f(x_2)}{\omega_3'(x_2)}.$$

下面我们利用恒等式 $L_n(x)=N_n(x)$ 证明性质 1 对任意的正整数 k 成立. 事实上

$$L_n(x)=\sum_{k=0}^{n}\frac{\omega_{n+1}(x)}{(x-x_k)\omega'_{n+1}(x_k)}y_k$$
$$=\sum_{k=0}^{n}\frac{y_k}{\omega'_{n+1}(x_k)}\frac{\omega_{n+1}(x)}{x-x_k},$$

即

$$L_n(x)=\frac{y_0}{\omega'_{n+1}(x_0)}\frac{\omega_{n+1}(x)}{x-x_0}+\frac{y_1}{\omega'_{n+1}(x_1)}\frac{\omega_{n+1}(x)}{x-x_1}+\cdots+\frac{y_n}{\omega'_{n+1}(x_n)}\frac{\omega_{n+1}(x)}{x-x_n}.$$

故 $L_n(x)$ 的首项系数应为

$$\frac{y_0}{\omega'_{n+1}(x_0)}+\frac{y_1}{\omega'_{n+1}(x_1)}+\cdots+\frac{y_n}{\omega'_{n+1}(x_n)}=\sum_{k=0}^{n}\frac{y_k}{\omega'_{n+1}(x_k)}=\sum_{k=0}^{n}\frac{f(x_k)}{\omega'_{n+1}(x_k)}.$$

而对于
$$N_n(x)=\sum_{k=0}^{n}f[x_0,x_1,\cdots,x_k]\omega_k(x),$$

它的首项应为

$$f[x_0,x_1,\cdots,x_n](x-x_0)\cdots(x-x_{n-1}),$$

其系数为
$$f[x_0,x_1,\cdots,x_n].$$

因为 $L_n(x)\equiv N_n(x)$,故它们的首项系数应相等. 所以有

$$f[x_0,x_1,\cdots,x_n]=\sum_{k=0}^{n}\frac{f(x_k)}{\omega'_{n+1}(x_k)}.$$

同理
$$f[x_0,x_1,\cdots,x_k]=\sum_{j=0}^{k}\frac{f(x_j)}{\omega'_{n+1}(x_j)}.$$

(2)均差的对称性,即

$$f[x_0,\cdots,x_i,\cdots,x_j,\cdots,x_k]=f[x_0,\cdots,x_j,\cdots,x_i,\cdots,x_k].$$

证 由性质 1,即式(6-11),

$$f[x_0,\cdots,x_i,\cdots,x_j,\cdots,x_k]=\frac{f(x_0)}{\omega'_{k+1}(x_0)}+\cdots+\frac{f(x_i)}{\omega'_{k+1}(x_i)}+\cdots$$
$$+\frac{f(x_j)}{\omega'_{k+1}(x_j)}+\cdots+\frac{f(x_k)}{\omega'_{k+1}(x_k)}$$
$$=f[x_0,\cdots,x_j,\cdots,x_i,\cdots,x_k],$$

再由对称性可得 k 阶均差的另一种计算公式:

$$f[x_0,x_1,\cdots,x_{k-2},x_{k-1},x_k]=\frac{f[x_0,x_1,\cdots,x_{k-2},x_k]-f[x_0,x_1,\cdots,x_{k-2},x_{k-1}]}{x_k-x_{k-1}}.$$

事实上

$$f[x_0,x_1,\cdots,x_{k-1},x_k]=f[x_{k-1},x_0,x_1,\cdots,x_{k-2},x_k]$$
$$=\frac{f[x_0,x_1,\cdots x_{k-2},x_k]-f[x_{k-1},x_0,\cdots,x_{k-2}]}{x_k-x_{k-1}}$$
$$=\frac{f[x_0,x_1,\cdots,x_{k-2},x_k]-f[x_0,x_1,\cdots,x_{k-2},x_{k-1}]}{x_k-x_{k-1}}.$$

(3)
$$f[x_0,x_1,\cdots,x_k]=\frac{f^{(k)}(\xi)}{k!},\quad \xi\in[a,b].$$

证 设$(x_i,y_i)_{i=0,1,\cdots,k}$，令$N_k(x)$是由上述$k+1$个结点构成的$k$阶Newton插值多项式.

因为
$$N_k(x_i)=y_i,\quad i=0,1,\cdots,k,$$
所以
$$f(x_i)-N_k(x_i)=0,\quad i=0,1,\cdots,k.$$
故$f(x)-N_k(x)$具有$k+1$个零点.

由Roll定理，$f^{(k)}(x)-N_k^{(k)}(x)$至少有一个零点ξ，记为$\xi\in(a,b)$. 所以
$$f^{(k)}(\xi)-N_k^{(k)}(\xi)=0,$$
即
$$f^{(k)}(\xi)=N_k^{(k)}(\xi)=f[x_0,x_1,\cdots,x_k]k!$$
故
$$f[x_0,x_1,\cdots,x_k]=\frac{f^{(k)}(\xi)}{k!}.$$

6.4 埃尔米特(Hermite)插值方法

前面所讨论的代数插值问题只要求插值多项式$p_n(x)$满足插值条件(6-2)，这些条件仅对结点处的函数值作了约束，因而所得插值多项式不能全面反映被插值函数$f(x)$的性态. 如果插值条件再增加对结点处导数的限制，则所构造的多项式会更好地逼近函数$f(x)$.

6.4.1 Hermite插值多项式

设$(x_i,y_i,y_i')_{i=0,1,\cdots,n}$，其中
$$x_i\neq x_j(i\neq j),\quad y_i=f(x_i),\quad y_i'=f'(x_i).$$
求一个$2n+1$次多项式$H_{2n+1}(x)$满足插值条件
$$\begin{cases}H_{2n+1}(x_i)=y_i,\\ H_{2n+1}'(x_i)=y_i',\quad i=0,1,\cdots,n.\end{cases}\tag{6-12}$$

首先构造基函数：

设$\alpha_j(x),\beta_j(x)(j=0,1,\cdots,n)$均为$2n+1$次多项式，且满足以下条件
$$\begin{cases}\alpha_j(x_k)=\begin{cases}0, & k\neq j,\\ 1, & k=j,\end{cases}\\ \alpha_j'(x_k)=0,\quad j,k=0,1,\cdots,n;\end{cases}\tag{6-13}$$
$$\begin{cases}\beta_j(x_k)=0,\\ \beta_j'(x_k)=\begin{cases}0, & k\neq j,\\ 1, & k=j,\quad j,k=0,1,\cdots,n.\end{cases}\end{cases}\tag{6-14}$$
据式(6-13)，用待定系数法容易推导出$\alpha_j(x)$的表达式
$$\alpha_j(x)=\Big[1-2(x-x_j)\sum_{\substack{k=0\\k\neq j}}^{n}\frac{1}{x_j-x_k}\Big]l_j^2(x),\tag{6-15}$$
其中
$$l_j(x)=\prod_{\substack{k=0\\k\neq j}}^{n}\frac{x-x_k}{x_j-x_k},$$
据式(6-14)，用待定系数法同样容易推得

$$\beta_j(x)=(x-x_j)l_j^2(x). \tag{6-16}$$

由基函数式(6－15)及式(6－16)可得 Hermite 插值多项式的表达式

$$H_{2n+1}(x)=\sum_{j=0}^{n}[y_j\alpha_j(x)+y_j{}'\beta_j(x)]. \tag{6-17}$$

显然，$H_{2n+1}(x)$满足插值条件(6－12)，即

$$\begin{cases} H_{2n+1}(x_i)=\sum\limits_{j=0}^{n}[y_j\alpha_j(x_i)+y_j{}'\beta_j(x_i)]=y_i, \\ H'_{2n+1}(x_i)=\sum\limits_{j=0}^{n}[y_j\alpha_j{}'(x_i)+y_j{}'\beta_j{}'(x_i)]=y_i{}', \quad i=0,1,\cdots,n. \end{cases}$$

6.4.2 两点三次 Hermite 插值公式

设$(x_i,y_i,y_i{}')_{i=k,k+1}$，求三次多项式 $H_3(x)$满足插值条件

$$\begin{cases} H_3(x_i)=y_i, \\ H_3{}'(x_i)=y_i{}', \quad i=k,k+1. \end{cases}$$

基函数 $\alpha_j(x)$，$\beta_j(x)(j=k,k+1)$均为三次多项式，且满足

$$\begin{cases} \alpha_j(x_i)=\begin{cases}0, & i\neq j,\\ 1, & i=j,\end{cases} \\ \alpha_j{}'(x_i)=0, \quad j,i=k,k+1; \end{cases}$$

$$\begin{cases} \beta_j(x_i)=0, \\ \beta_j{}'(x_j)=\begin{cases}0, & i\neq j,\\ 1, & i=j, \quad j,i=k,k+1.\end{cases} \end{cases}$$

据式(6－15)及式(6－16)可得

$$\begin{cases} \alpha_k(x)=\left(1+2\dfrac{x-x_k}{x_{k+1}-x_k}\right)l_k^2(x), \\ \alpha_{k+1}(x)=\left(1+2\dfrac{x-x_{k+1}}{x_k-x_{k+1}}\right)l_{k+1}^2(x); \end{cases} \tag{6-18}$$

$$\begin{cases} \beta_k(x)=(x-x_k)l_k^2(x), \\ \beta_{k+1}(x)=(x-x_{k+1})l_{k+1}^2(x). \end{cases} \tag{6-19}$$

其中
$$l_k(x)=\frac{x-x_{k+1}}{x_k-x_{k+1}}, \quad l_{k+1}(x)=\frac{x-x_k}{x_{k+1}-x_k}.$$

所以，两点三次 Hermite 插值多项式的表达式为

$$H_3(x)=y_k\alpha_k(x)+y_{k+1}\alpha_{k+1}(x)+y_k{}'\beta_k(x)+y'_{k+1}\beta_{k+1}(x). \tag{6-20}$$

例 5 已知数据见表 6－8. 试确定三次多项式 $p_3(x)$满足

$$\begin{cases} p_3(x_i)=y_i, \quad i=0,1,2, \\ p_3{}'(x_1)=y_1. \end{cases}$$

表6-8

x_i	x_0	x_1	x_2
y_i	y_0	y_1	y_2
y_i'		y_1'	

这是一个非典型的 Hermite 插值问题(缺少导数条件),若令

$$p_3(x)=\alpha_0(x)y_0+\alpha_1(x)y_1+\alpha_2(x)y_2+\beta_1(x)y_1',$$

则因为缺少导数条件,不能直接利用式(6-15)及式(6-16)写出 $\alpha_j(x)(j=0,1,2)$及$\beta_1(x)$的表达式,但可根据式(6-13)及式(6-14)采用待定系数法确定 $\alpha_j(x)(j=0,1,2)$及$\beta_1(x)$的表达式. 这种解法太复杂也不易掌握. 下面介绍一种混合方法.

解 首先利用表 6-8 中数据$(x_i,y_i)_{i=0,1,2}$构造二次插值多项式 $N_2(x)$. 建立均差表(见表 6-9).

表6-9

x_i	$f(x_i)$	一阶均差	二阶均差
x_0	$f(x_0)=y_0$		
x_1	$f(x_1)=y_1$	$f[x_0,x_1]$	
x_2	$f(x_2)=y_2$	$f[x_1,x_2]$	$f[x_0,x_1,x_2]$

$$N_2(x)=f(x_0)+f[x_0,x_1](x-x_0)+f[x_0,x_1,x_2](x-x_0)(x-x_1).$$

令
$$p_3(x)=N_2(x)+Q_3(x).$$

因为
$$p_3(x_i)=N_2(x_i)=y_i,\quad i=0,1,2,$$

所以
$$Q_3(x_i)=p_3(x_i)-N_2(x_i)=0,\quad i=0,1,2.$$

故
$$Q_3(x)=a(x-x_0)(x-x_1)(x-x_2).$$

再据
$$p_3'(x_1)=y_1',$$

即有
$$f[x_0,x_1]+f[x_0,x_1,x_2](x_1-x_0)+a(x_1-x_0)(x_1-x_2)=y_1',$$

解得
$$a=\frac{y_1'-f[x_0,x_1]-f[x_0,x_1,x_2](x_1-x_0)}{(x_1-x_0)(x_1-x_2)}.$$

所以

$$p_3(x)=f(x_0)+f[x_0,x_1](x-x_0)+f[x_0,x_1,x_2](x-x_0)(x-x_1)$$
$$+a(x-x_0)(x-x_1)(x-x_2).$$

例 6 已知数据如表 6-10 所示. 求 $p_4(x)$满足插值条件

$$p_4(x_i)=y_i,\quad i=0,1,2;$$
$$p_4'(x_i)=y_i',\quad i=0,1.$$

表6-10

x_i	$x_0=0$	$x_1=1$	$x_2=2$
y_i	$y_0=0$	$y_1=1$	$y_2=1$
y_i'	$y_0'=0$	$y_1'=1$	

解 首先根据表 6-10 中数据 $(x_i, y_i)_{i=0,1,2}$ 建立均差表得到二次插值多项式

$$N_2(x) = -\frac{1}{2}x^2 + \frac{3}{2}x.$$

令

$$p_4(x) = N_2(x) + Q_4(x).$$

因为

$$Q_4(x_i) = p_4(x_i) - N_2(x_i) = 0, \quad i = 0,1,2,$$

所以

$$Q_4(x) = (ax+b)x(x-1)(x-2).$$

再利用

$$p_4'(x_i) = y_i', \quad i = 0,1,$$

得

$$\begin{cases} p_4'(0) = 0, \\ p_4'(1) = 1, \end{cases}$$

即

$$\begin{cases} \frac{3}{2} + 2b = 0, \\ a + b = -\frac{1}{2}. \end{cases}$$

最后解得

$$a = \frac{1}{4}, \quad b = -\frac{3}{4}.$$

所以

$$p_4(x) = -\frac{1}{2}x^2 + \frac{3}{2}x + \frac{1}{4}(x-3)x(x-1)(x-2)$$

$$= \frac{1}{2}x(3-x) + \frac{1}{4}x(x-1)(x-2)(x-3).$$

6.4.3 分段低阶插值

例 7 龙格(Runge)振荡现象.

设

$$f(x) = \frac{1}{1+25x^2}, \quad x \in [-1,1],$$

在区间[−1,1]上取等距节点 $x_i = -1 + \frac{2}{10}i\,(i = 0,1,\cdots,10)$，利用节点 $(x_i, f(x_i))$ $i = 0,1,\cdots,10$ 作 Lagrange 插值多项式 $L_{10}(x)$，即

$$L_{10}(x) = \sum_{i=0}^{10} \frac{1}{1+25x_i^2} l_i(x),$$

其中

$$l_i(x) = \prod_{\substack{j=0 \\ j\neq i}}^{10} \frac{x - x_j}{x_i - x_j}.$$

将 $L_{10}(x)$ 在区间[0,1]上的计算结果及图形分别示于表 6−11 和图 6−1.

表 6−11

x_i	$f(x_i)$	$L_{10}(x_i)$
0.00	1.000 00	1.000 00
0.10	0.800 00	0.843 40
0.20	0.500 00	0.500 00
0.30	0.377 69	0.235 35
0.40	0.200 00	0.199 99
0.50	0.137 93	0.253 76

续表 6－11

x_i	$f(x_i)$	$L_{10}(x_i)$
0.60	0.100 00	0.100 00
0.70	0.075 47	−0.226 20
0.80	0.058 82	0.058 82
0.90	0.047 06	1.578 72
1.00	0.038 46	0.038 46

从图 6－1 及表 6－11 可以看出，在闭区间$[-1,1]$上，只有在以原点为中心的一个小区间内，$f(x)$的值与$L_{10}(x)$的值比较接近，在此区间之外，$L_{10}(x)$的图形振荡得很厉害．我们还可以证明，当$n\to\infty$时，$L_n(x)$只能在$|x|<0.363$内与$f(x)$逼近，而在此区间之外是发散的，这一现象称为高次插值多项式的 Runge 振荡现象．为了克服上述困难，通常采用分段低阶插值方法，例如采用分段线性插值、分段二次插值、分段三次插值（两点三次 Hermite 插值）、三次样条插值等．

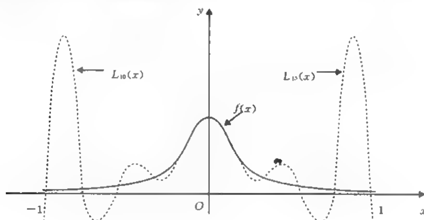


图 6－1

6.5 三次样条插值方法

样条（Spline）插值，是一种既能克服高次插值的缺陷，又能保证函数曲线具有一定光滑性的插值，即是一种具有一定光滑性的分段插值．

样条一词来源于工程中的样条曲线．绘图员在描绘过已知点（样点）的光滑曲线时，往往用细长的木条或金属条（样条）把邻近的几个点连接成光滑曲线．这一过程，就力学而言，相当于细梁（样条）在若干点处受到集中载荷的挠曲．这样绘出的曲线实际上是一段段三次曲线拼成的曲线．在拼接处，函数不仅连续，而且它的一阶及二阶导数也连续．通常把这样得到的函数叫作样条函数．

6.5.1 三次样条插值函数

设$(x_i, y_i)_{i=0,1,\cdots,n}$，其中 $y_i=f(x_i)$，且结点的分布满足

$$a=x_0<x_1<\cdots<x_n=b.$$

若 $s(x)$满足：

(1)连续条件，即 $s(x), s'(x), s''(x)$在$[a,b]$上连续；

(2)分段多项式

$$s_i(x)=a_i+b_i x+c_i x^2+d_i x^3;$$

(3)插值条件，即

$$s(x_j)=y_j, \quad j=0,1,\cdots,n;$$

我们则称 $s(x)$为在区间$[a,b]$上的三次样条插值函数. 它是区间$[a,b]$上的分段函数，即

$$s(x)=\begin{cases} s_1(x), & x\in[x_0,x_1], \\ s_2(x), & x\in[x_1,x_2], \\ \quad\vdots \\ s_n(x), & x\in[x_{n-1},x_n]. \end{cases}$$

其中 $s_i(x), x\in[x_{i-1},x_i], i=1,2,\cdots,n.$

6.5.2 三次样条插值函数的构成

1)利用插值条件构造 $s(x)$在子区间上的表达式

$s(x)$在子区间上的表达式 $s_i(x)(i=1,2,\cdots,n)$如图 6－2 所示.

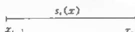


图 6－2

已知 $s_i(x_{i-1})=y_{i-1}, s_i(x_i)=y_i$，设

$$s_i'(x_{i-1})=m_{i-1}, \quad s_i'(x_i)=m_i.$$

则由式(6－20)得

$$s_i(x)=\alpha_{i-1}(x)y_{i-1}+\alpha_i(x)y_i+\beta_{i-1}(x)m_{i-1}+\beta_i(x)m_i. \tag{6－21}$$

其中 $\alpha_{i-1}(x), \alpha_i(x), \beta_{i-1}(x), \beta_i(x)$分别由式(6－18)及式(6－19)确定，即

$$\begin{cases} \alpha_{i-1}(x)=\left(1+2\dfrac{x-x_{i-1}}{x_i-x_{i-1}}\right)\left(\dfrac{x-x_i}{x_{i-1}-x_i}\right)^2, \\ \alpha_i(x)=\left(1+2\dfrac{x-x_i}{x_{i-1}-x_i}\right)\left(\dfrac{x-x_{i-1}}{x_i-x_{i-1}}\right)^2; \end{cases}$$

$$\begin{cases} \beta_{i-1}(x)=(x-x_{i-1})\left(\dfrac{x-x_i}{x_{i-1}-x_i}\right)^2, \\ \beta_i(x)=(x-x_i)\left(\dfrac{x-x_{i-1}}{x_i-x_{i-1}}\right)^2. \end{cases}$$

值得注意的是，在式(6－21)中，$n+1$ 个未知数 $m_i(i=0,1,\cdots,n)$尚待确定.

2)利用连续条件建立关于 $m_i(i=0,1,\cdots,n)$ 的线性方程组

如图 6－3 所示，在区间 $[a,b]$ 上共有 $n-1$ 个内点 $x_i(i=1,2,\cdots,n-1)$. $s''(x)$ 在内点 x_i 处连续，即是

$$s_i''(x_i^-)=s_{i+1}''(x_i^+),$$

亦即

$$\lim_{x\to x_i^-} s_i''(x)=\lim_{x\to x_i^+} s_{i+1}''(x).$$



图 6－3

令 $h_{i-1}=x_i-x_{i-1}$，计算 $s_i''(x)$，即

$$s_i''(x)=\frac{6x-2x_{i-1}-4x_i}{h_{i-1}^2}m_{i-1}+\frac{6x-4x_{i-1}-2x_i}{h_{i-1}^2}m_i$$

$$+\frac{6(x_{i-1}+x_i-2x)}{h_{i-1}^3}(y_i-y_{i-1}), \tag{6-22}$$

$$s_i''(x_i^-)=\lim_{x\to x_i^-}s_i''(x)=\frac{2}{h_{i-1}}m_{i-1}+\frac{4}{h_{i-1}}m_i-\frac{6}{h_{i-1}}(y_i-y_{i-1}).$$

令 $h_i=x_{i+1}-x_i$，计算 $s_{i+1}''(x)$，即

$$s_{i+1}''(x)=\frac{6x-2x_i-4x_{i+1}}{h_i^2}m_i+\frac{6x-4x_i-2x_{i+1}}{h_i^2}m_{i+1}$$

$$+\frac{6(x_i+x_{i+1}-2x)}{h_i^3}(y_{i+1}-y_i), \tag{6-23}$$

$$s_{i+1}''(x_i^+)=\lim_{x\to x_i^+}s_{i+1}''(x)=-\frac{4}{h_i}m_i-\frac{2}{h_i}m_{i+1}+\frac{6}{h_i}(y_{i+1}-y_i).$$

令 $s_i''(x_i^-)=s_{i+1}''(x_i^+)$，整理得

$$\mu_i m_{i-1}+2m_i+\lambda_i m_{i+1}=d_i,\quad i=1,2,\cdots,n-1. \tag{6-24}$$

其中

$$\mu_i=\frac{h_i}{h_{i-1}+h_i},\quad \lambda_i=\frac{h_{i-1}}{h_{i-1}+h_i},$$

$$d_i=3(\mu_i f[x_{i-1},x_i]+\lambda_i f[x_i,x_{i+1}]).$$

在式(6－24)中，共有 $n-1$ 个线性方程，$n+1$ 个未知数 $m_i(i=0,1,\cdots,n)$，因此，欲求解方程组(6－24)，必须补充条件.

3)补充边界条件

(1)第一边界条件：

设 $\qquad m_0=s_1'(x_0)=y_0',\quad m_n=s_n'(x_n)=y_n',$

并将其代入式(6－24)，得到 $n-1$ 阶三对角方程组

$$
\begin{cases}
2m_1+\lambda_1 m_2=d_1-\mu_1 y_0', \\
\mu_2 m_1+2m_2+\lambda_2 m_3=d_2, \\
\qquad\vdots \\
\mu_i m_{i-1}+2m_i+\lambda_i m_{i+1}=d_i, \\
\qquad\vdots \\
\mu_{n-2}m_{n-3}+2m_{n-2}+\lambda_{n-2}m_{n-1}=d_{n-2}, \\
\mu_{n-1}m_{n-2}+2m_{n-1}=d_{n-1}-\lambda_{n-1}y_n'.
\end{cases}
\tag{6-25}
$$

将三对角方程组(6－25)写成矩阵形式 $\boldsymbol{Ax}=\boldsymbol{b}$，其中

$$
\boldsymbol{A}=\begin{bmatrix}
2 & \lambda_1 & & & & & \\
\mu_2 & 2 & \lambda_2 & & & & \\
 & \ddots & \ddots & \ddots & & & \\
 & & \mu_i & 2 & \lambda_i & & \\
 & & & \ddots & \ddots & \ddots & \\
 & & & & \mu_{n-2} & 2 & \lambda_{n-2} \\
 & & & & & \mu_{n-1} & 2
\end{bmatrix},
$$

$$
\boldsymbol{x}=(m_1,m_2,\cdots,m_{n-1})^{\mathrm{T}},
$$

$$
\boldsymbol{b}=(d_1-\mu_1 y_0',d_2,\cdots,d_{n-2},d_{n-1}-\lambda_{n-1}y_n')^{\mathrm{T}}.
$$

(2)第二边界条件：

设 $$s_1''(x_0)=0,\quad s_n''(x_n)=0.$$

据式(6－22)，$s_1''(x_0)=0$，即有

$$
2m_0+m_1=\frac{3}{h_0}(y_1-y_0);
$$

据式(6－22)，$s_n''(x_n)=0$，即有

$$
m_{n-1}+2m_n=\frac{3}{h_{n-1}}(y_n-y_{n-1}).
$$

将上述两个方程与式(6－24)联立得到 $n+1$ 阶三对角方程组

$$
\boldsymbol{Cx}=\boldsymbol{f}, \tag{6-26}
$$

其中

$$
\boldsymbol{C}=\begin{bmatrix}
2 & 1 & & & & & \\
\mu_2 & 2 & \lambda_2 & & & & \\
 & \ddots & \ddots & \ddots & & & \\
 & & \mu_i & 2 & \lambda_i & & \\
 & & & \ddots & \ddots & \ddots & \\
 & & & & \mu_{n-1} & 2 & \lambda_{n-1} \\
 & & & & & 1 & 2
\end{bmatrix},
$$

$$
\boldsymbol{x}=(m_0,m_1,\cdots,m_n)^{\mathrm{T}},
$$

$$
\boldsymbol{f}=\left[\frac{3}{h_0}(y_1-y_0),d_1,\cdots,d_{n-1},\frac{3}{h_{n-1}}(y_n-y_{n-1})\right]^{\mathrm{T}}.
$$

6.5.3 第一边界条件下样条插值算法

Algo. 6.1

1° 送初值 $x_i, y_i (i=0,1,\cdots,n), y_0', y_n', n$.

2° 计算步长

$$h_i = x_{i+1} - x_i (i=0,1,\cdots,n-1); \quad h_{i-1} = x_i - x_{i-1} (i=1,2,\cdots,n).$$

3° 计算均差

$$f[x_i, x_{i+1}] = \frac{y_{i+1} - y_i}{h_i}, \quad i=0,1,\cdots,n-1;$$

$$f[x_{i-1}, x_i] = \frac{y_i - y_{i-1}}{h_{i-1}}, \quad i=1,2,\cdots,n.$$

4° 计算 $\mu_i, \lambda_i, d_i (i=1,2,\cdots,n-1)$，即

$$\mu_i = \frac{h_i}{h_{i-1}+h_i}, \quad \lambda_i = \frac{h_{i-1}}{h_{i-1}+h_i},$$

$$d_i = 3(\mu_i f[x_{i-1}, x_i] + \lambda_i f[x_i, x_{i+1}]).$$

5° 解三对角方程组(6-25)得 $m_i (i=1,2,\cdots,n-1)$.

6° 计算表达式

$$s_i(x) = \alpha_{i-1}(x) y_{i-1} + \alpha_i(x) y_i + \beta_{i-1}(x) m_{i-1} + \beta_i(x) m_i, \quad i=1,2,\cdots,n,$$

其中 $\alpha_{i-1}(x), \alpha_i(x), \beta_{i-1}(x), \beta_i(x)$ 按式(6-20)中基函数的计算公式计算.

7° 对给定的插值点 x，计算 $s(x)$.

8° 输出 $s(x)$.

例 8 给定数据如表 6-12 所示. 试根据第一、第二边界条件

$$s'(0.25)=1, \quad s'(0.53)=0.6868,$$

$$s''(0.25)=s''(0.53)=0,$$

分别计算三次样条函数 $s(x)$.

表 6-12

x_i	0.25	0.30	0.39	0.45	0.53
y_i	0.5000	0.5477	0.6245	0.6708	0.7280

解 (1)首先，根据给定数据按 Algo. 6.1 计算中间数据示于表 6-13.

表 6-13

i	x_i	y_i	h_i	μ_i	λ_i	d_i
0	0.25	0.5000	0.05	—	—	—
1	0.30	0.5477	0.09	0.6427	0.3571	2.7542
2	0.39	0.6245	0.06	0.4	0.6	2.4130

续表 6-13

i	x_i	y_i	h_i	μ_i	λ_i	d_i
3	0.45	0.670 8	0.08	0.571 4	0.428 6	2.242 1
4	0.53	0.728 0		—	—	—

其次，据式(6-25)写出三阶三对角方程组

$$\begin{bmatrix} 2 & \lambda_1 & 0 \\ \mu_2 & 2 & \lambda_2 \\ 0 & \mu_3 & 2 \end{bmatrix}\begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix}=\begin{bmatrix} d_1-\mu_1 y_0' \\ d_2 \\ d_3-\lambda_3 y_4' \end{bmatrix},$$

即

$$\begin{bmatrix} 2 & 0.357\,1 & 0 \\ 0.4 & 2 & 0.6 \\ 0 & 0.571\,4 & 2 \end{bmatrix}\begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix}=\begin{bmatrix} 2.111\,3 \\ 2.413\,0 \\ 1.947\,1 \end{bmatrix}.$$

从而解得

$$\boldsymbol{m}=(m_1,m_2,m_3)^{\mathrm{T}}=(0.846\,8,0.585\,8,0.745\,2)^{\mathrm{T}}.$$

最后，将 $m_0=1,m_4=0.686\,8,m_i(i=1,2,3)$代入式(6-21)得

$$s_i(x)=\alpha_{i-1}(x)y_{i-1}+\alpha_i(x)y_i+\beta_{i-1}(x)m_{i-1}+\beta_i(x)m_i,\quad i=1,2,3,4.$$

其中，$s_i(x)(i=1,2,3,4)$的具体表达式见习题详解.

(2)设 $s''(0.25)=s''(0.53)=0.$

根据中间数据表 6-13 及方程组(6-26)写出五阶三对角方程组

$$\begin{bmatrix} 2 & 1 & 0 & 0 & 0 \\ \mu_1 & 2 & \lambda_1 & 0 & 0 \\ 0 & \mu_2 & 2 & \lambda_2 & 0 \\ 0 & 0 & \mu_3 & 2 & \lambda_3 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix}\begin{bmatrix} m_0 \\ m_1 \\ m_2 \\ m_3 \\ m_4 \end{bmatrix}=\begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix},$$

其中

$$d_0=\frac{3}{h_0}(y_1-y_0)=2.862,$$

$$d_4=\frac{3}{h_3}(y_4-y_3)=2.145.$$

即

$$\begin{bmatrix} 2 & 1 & 0 & 0 & 0 \\ 0.642\,9 & 2 & 0.357\,1 & 0 & 0 \\ 0 & 0.4 & 2 & 0.6 & 0 \\ 0 & 0 & 0.571\,4 & 2 & 0.428\,6 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix}\begin{bmatrix} m_0 \\ m_1 \\ m_2 \\ m_3 \\ m_4 \end{bmatrix}=\begin{bmatrix} 2.862 \\ 2.754\,2 \\ 2.413\,6 \\ 2.242\,1 \\ 2.145 \end{bmatrix},$$

从而解得

$$\boldsymbol{m}=(0.969\,7,0.922\,7,0.799\,3,0.742\,4,0.701\,3)^{\mathrm{T}}.$$

将 $m_i(i=0,1,2,3,4)$代入式(6-21)得

$$s_i(x)=\alpha_{i-1}(x)y_{i-1}+\alpha_i(x)y_i+\beta_{i-1}(x)m_{i-1}+\beta_i(x)m_i,\quad i=1,2,3,4.$$

其中，$s_i(x)(i=1,2,3,4)$的具体表达式见习题详解.

6.6 曲线拟合

6.6.1 问题的提出

给定实测数据$(x_i,y_i)_{i=1,2,\cdots,m}$，其中

$$x_i\neq x_j(i\neq j),\quad 且\ y_i=f(x_i),\quad i=1,2,\cdots,m.$$

插值方法(多项式插值)：

寻求一个次数不超过 $m-1$ 次的多项式 $p_{m-1}(x)$(如图 6－3)满足插值条件

$$p_{m-1}(x_i)=y_i,\quad i=1,2,\cdots,m;$$

拟合方法(多项式拟合)：

寻求一个次数不超过 n 次的多项式 $p_n(x)$，$n\ll m$(如图 6－4)，要求偏差平方和达到最小，即

$$\left[\sum_{i=1}^{m}(p_n(x_i)-y_i)^2\right]_{\min}.$$

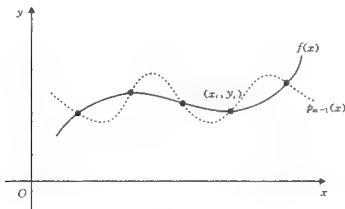


图 6－4

在图 6－4 及图 6－5 中，$f(x)$代表实测数据点(图中黑点)所对应的函数曲线(实线)，$p_{m-1}(x)$(虚线)代表插值曲线，$p_n(x)$(虚线)代表拟合曲线.

从这两个图形看出，插值曲线必须通过所给实测点(x_i,y_i)，而拟合曲线不必通过(x_i,y_i)，只要求偏差平方和达到最小. 因此，插值曲线 $p_{m-1}(x)$只反映 $f(x)$的局部性质，而拟合曲线 $p_n(x)$则反映了数据(x_i,y_i)的总体趋势.

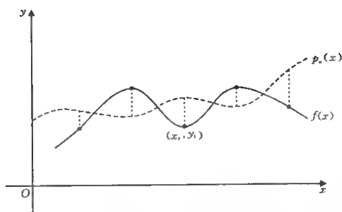


图 6－5

6.6.2 实例分析

例 9 摩尔(Moore)预测公式.

Intel 公司董事长 Moore 在上个世纪的 60 年代就观察到一个很有趣的现象:集成电路上可容纳的单晶体数量每隔一年半(18 个月)左右便会增长一倍,从而使集成电路的性能也能提高一倍.据此他提出了轰动世界的 Moore 定律,预测这种增长趋势会一直延续下去.

下面我们给出 Moore 数据,如表 6－14 所示.

表 6 14

t_i(年)	1 959	1 962	1 963	1 964	1 965
k_i(增长倍数)	1	3	4	5	6

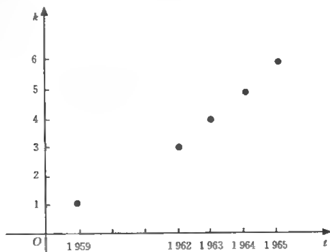


图 6 6

表 6－14 中第二行数据为芯片上晶体数目在不同年代与 1 959 年时的数目比较的倍数. 经观察 k 与 t 之间呈拟线性关系，如图 6－6. 据此 Moore 导出了著名的经验公式 Moore 定律.

解 上述问题不宜用插值方法去解决，因为若用到全部五组数据则需构造 4 次多项式，这与数据的拟线性趋势不合；若用两组数据构造线性插值式又不能反映 k 与 t 的真实关系. 为此可设

$$K(t)=a+bt, \tag{6-27}$$

将表 6－14 中的数据代入式(6－27)得超定方程组

$$k_i=a+bt_i,\quad i=1,2,3,4,5.$$

其中，t 表示时间，k 表示增长倍数，a,b 为待定参数.

若将表 6－14 中的数据代入式(6－27)，得线性方程组

$$\begin{cases} a+1\,959b=1, \\ a+1\,962b=3, \\ a+1\,963b=4, \\ a+1\,964b=5, \\ a+1\,965b=6. \end{cases} \tag{6-28}$$

方程组(6－28)是一个超定方程组. 在这五个线性方程中，任意两个联立求解可得到十组不同的解，即是说该方程组不存在通常义上的解.

现将线性方程组(6－28)写成矩阵形式 $\boldsymbol{Ax}=\boldsymbol{y}$，其中

$$\boldsymbol{A}=\begin{bmatrix} 1 & 1\,959 \\ 1 & 1\,962 \\ 1 & 1\,963 \\ 1 & 1\,964 \\ 1 & 1\,965 \end{bmatrix},\quad \boldsymbol{x}=(a,b)^{\mathrm{T}},\quad \boldsymbol{y}=(1,3,4,5,6)^{\mathrm{T}}.$$

此超定方程组无常义解，即是说不存在 $\boldsymbol{x}^*\in R^2$ 使得 $\boldsymbol{Ax}^*=\boldsymbol{y}$，但是我们将在 6.6.3 中证明该超定方程组存在最小二乘解，也即是说存在 $\boldsymbol{x}^*\in R^2$，使得 $\|\boldsymbol{Ax}^*-\boldsymbol{y}\|_2$ 达到最小，并且 $\boldsymbol{x}^*$ 是线性方程组

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{Ax}^*=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} \tag{6-29}$$

的解. 我们称式(6－29)为法方程组，在本例中它是一个二阶线性方程组，即

$$\begin{bmatrix} 5 & 9\,813 \\ 9\,813 & 19\,259\,015 \end{bmatrix}\begin{bmatrix} a^* \\ b^* \end{bmatrix}=\begin{bmatrix} 19 \\ 37\,307 \end{bmatrix}.$$

解这个方程组得

$$\boldsymbol{x}^*=(a^*,b^*)^{\mathrm{T}}=(-1\,625.550\,3,0.830\,2)^{\mathrm{T}}.$$

由此得到 Moore 公式

$$K(t)=-1\,625.550\,3+0.830\,2t.$$

对于 $K(t)$，显然 $K(t_i)\neq k_i(i=1,2,3,4,5)$，但是 $\sum\limits_{i=1}^{5}[K(t_i)-k_i]^2=0.188\,7$(此值很小).

将上述问题的求解步骤归纳如下：

(1)据散点图 6－6 设立拟合函数

$$K(t)=a+bt.$$

(2)代入表 6－14 中的数据得超定方程组

$$a+bt_i=k_i,\quad i=1,2,3,4,5,$$

该超定方程组的矩阵形式为 $\boldsymbol{Ax}=\boldsymbol{y}$，其中

$$\boldsymbol{A}=\begin{bmatrix}1 & t_1\\1 & t_2\\1 & t_3\\1 & t_4\\1 & t_5\end{bmatrix}=\begin{bmatrix}1 & 1\,959\\1 & 1\,962\\1 & 1\,963\\1 & 1\,964\\1 & 1\,965\end{bmatrix},$$

$$\boldsymbol{x}=(a,b)^{\mathrm{T}},\quad \boldsymbol{y}=(k_1,k_2,k_3,k_4,k_5)^{\mathrm{T}}=(1,3,4,5,6)^{\mathrm{T}}.$$

(3)列表 6－15，建立法方程组 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{Ax}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y}$.

表 6－15

t_i	k_i	t_ik_i	t_i^2
1 959	1	1 959	3 837 681
1 962	3	5 886	3 849 444
1 963	4	7 852	3 853 369
1 964	5	9 820	3 857 296
1 965	6	11 790	3 861 225
$\sum_{i=1}^{5} t_i=9\,813$	$\sum_{i=1}^{5} k_i=19$	$\sum_{i=1}^{5} t_ik_i=37\,307$	$\sum_{i=1}^{5} t_i^2=19\,259\,015$

据表 6－15 中计算结果得

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}=\begin{bmatrix}m & \sum_{i=1}^{5}t_i\\ \sum_{i=1}^{5}t_i & \sum_{i=1}^{5}t_i^2\end{bmatrix}=\begin{bmatrix}5 & 9\,813\\9\,813 & 19\,259\,015\end{bmatrix},$$

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y}=\left(\sum k_i,\sum t_ik_i\right)^{\mathrm{T}}=(19,373\,07)^{\mathrm{T}}.$$

其中 m 为实测数据的组数.

(4)解法方程组得拟合参数向量

$$\boldsymbol{x}^*=(a^*,b^*)^{\mathrm{T}}=(-1\,625.550\,3,0.830\,2)^{\mathrm{T}},$$

并由此得 Moore 预测公式

$$K(t)=-1\,625.550\,3+0.830\,2t.$$

例 10 观测一个做直线运动的物体，测得该物体运动所需的时间 t_i 与路程 s_i 的关系如表 6－16 所示. 若物体的加速度为常数，试求该物体的初速度 v_0 和加速度 a.

表 6－16

t_i(s)	0	0.9	1.9	3.0	3.9	5.0
s_i(m)	0	10	30	50	80	110

解 (1)由题设知，该物体做匀加速运动(加速度为常数)，其运动方程为

$$s(t)=s_0+v_0t+\frac{1}{2}at^2$$
$$=v_0t+\frac{1}{2}at^2.$$

(2)将表 6 16 中的数据代入上式得超定方程组

$$v_0t_i+\frac{a}{2}t_i^2=s_i,\quad i=1,2,3,4,5.$$

该超定方程组的矩阵形式为 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{s}$，其中

$$\boldsymbol{A}=\begin{bmatrix} t_1 & t_1^2 \\ t_2 & t_2^2 \\ t_3 & t_3^2 \\ t_4 & t_4^2 \\ t_5 & t_5^2 \end{bmatrix}=\begin{bmatrix} 0.9 & 0.81 \\ 1.9 & 3.61 \\ 3.0 & 9 \\ 3.9 & 15.21 \\ 5.0 & 25 \end{bmatrix},$$

$$\boldsymbol{x}=(v_0,a/2)^{\mathrm{T}},\quad \boldsymbol{s}=(10,30,50,80,110)^{\mathrm{T}}.$$

(3)列表 6－17，建立法方程组

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{s}.$$

表 6－17

t_i	s_i	t_is_i	t_i^2	$t_i^2s_i$	t_i^3	t_i^4
0.9	10	9	0.81	8.1	0.729	0.656 1
1.9	30	57	3.61	108.3	6.859	13.032 1
3.0	50	150	9	450	27	81
3.9	80	312	15.21	1 216.8	59.319	231.344 1
5.0	110	550	25	275 0	125	625
—	—	$\sum_{i=1}^{5}t_is_i=1\,078$	$\sum_{i=1}^{5}t_i^2=53.63$	$\sum_{i=1}^{5}t_i^2s_i=4\,533.2$	$\sum_{i=1}^{5}t_i^3=218.907$	$\sum_{i=1}^{5}t_i^4=951.032\,3$

据表 6－12 中计算结果得

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}=\begin{bmatrix} \sum_{i=1}^{5}t_i^2 & \sum_{i=1}^{5}t_i^3 \\ \sum_{i=1}^{5}t_i^3 & \sum_{i=1}^{5}t_i^4 \end{bmatrix}=\begin{bmatrix} 53.63 & 218.907 \\ 218.907 & 951.032\,3 \end{bmatrix},$$

$$A^{T}s=(\sum_{i=1}^{5}t_is_i,\sum_{i=1}^{5}t_i^2s_i)^{T}=(1\,078,4\,533.2)^{T}.$$

(4)解法方程组 $A^{T}Ax=A^{T}s$,得

$$x^*=(v_0,a/2)^{T}=(10.657\,4,2.313\,5)^{T}.$$

所以该物体的初速度为 10.657 4 m/s,加速度为 4.627 m/s^2.

例 11　人口预测问题:由 20 世纪 60 年代世界人口统计数据(见表 6－18)估算 2000 年世界人口总数.

表 6－18

时间(年)	1 960	1 961	1 962	1 963	1 964	1 965	1 966	1 967	1 968
人口数($\times 10^6$)	2 972	3 061	3 151	3 213	3 234	3 285	3 356	3 420	3 483

对于这个问题曾有人根据表中数据,预测出 2000 年全世界人口会超过 60 亿.这一结论在 20 世纪 60 年代似乎危言耸听,但据 1999 年 12 月 10 日联合国发布的世界人口统计数据,全世界人口已达到 60 亿.

分析　据人口增长的统计资料和人口模型理论,当人口总数不是很大时,在不太长的时间内,人口增长接近于指数曲线.

解　(1)设拟合函数

$$N(t)=e^{a+bt},$$

其中,t 表示时间,$N(t)$表示人口数量.

将上式两端取对数得

$$\ln N(t)=a+bt.$$

令 $Z(t)=\ln N(t)$,得线性函数

$$Z(t)=a+bt,$$

并构造数据列于表 6－19 中.

表 6－19

t_i	1 960	1 961	1 962	1 963	1 964	1 965	1 966	1 967	1 968
$Z_i(\times 10^6)$	7.997 0	8.026 5	8.055 5	8.075 0	8.081 5	8.097 1	8.118 5	8.137 4	8.155 6

(2)将表 6－19 中的数据代入 $Z(t)=a+bt$,得超定方程组

$$a+bt_i=z_i,\quad i=1,2,\cdots,9,$$

其矩阵形式为 $Ax=Z$,其中

$$
\boldsymbol{A}=\begin{bmatrix}1 & 1\,960\\ 1 & 1\,961\\ 1 & 1\,962\\ 1 & 1\,963\\ 1 & 1\,964\\ 1 & 1\,965\\ 1 & 1\,966\\ 1 & 1\,967\\ 1 & 1\,968\end{bmatrix},\quad \boldsymbol{x}=(a,b)^{\mathrm{T}},
$$

$\boldsymbol{Z}=(7.997\,0, 8.026\,5, 8.055\,5, 8.075\,0, 8.081\,5, 8.097\,1, 8.118\,5, 8.137\,4, 8.155\,6)^{\mathrm{T}}$.

(3)列表 6－20，建立法方程组 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{Z}$.

表 6－20

t_i	z_i	$t_i z_i$	t_i^2
1 960	7.997 0	15 674.120 0	3 841 600
1 961	8.026 5	15 739.966 5	3 845 521
1 962	8.055 5	15 804.891 0	3 849 444
1 963	8.075 0	15 851.225 0	3 853 369
1 964	8.081 5	15 872.066 0	3 857 296
1 965	8.097 1	15 910.801 5	3 861 225
1 966	8.118 5	15 960.971 0	3 865 156
1 967	8.137 4	16 006.265 8	3 869 089
1 968	8.155 6	16 050.220 8	3 873 024
$\sum_{i=1}^{9} t_i=17\,676$	$\sum_{i=1}^{9} z_i=72.746\,8$	$\sum_{i=1}^{9} t_i z_i=142\,870.527\,6$	$\sum_{i=1}^{9} t_i^2=34\,715\,724$

据表 6－20 的计算结果得

$$
\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}=\begin{bmatrix} m & \sum_{i=1}^{9} t_i\\ \sum_{i=1}^{9} t_i & \sum_{i=1}^{9} t_i^2\end{bmatrix}=\begin{bmatrix}9 & 17\,676\\ 17\,676 & 34\,715\,724\end{bmatrix},
$$

$$
\boldsymbol{A}^{\mathrm{T}}\boldsymbol{Z}=\Big(\sum_{i=1}^{9} z_i,\sum_{i=1}^{9} t_i z_i\Big)^{\mathrm{T}}=(72.746\,8, 142\,870.527\,6)^{\mathrm{T}}.
$$

(4)解法方程得

$$
a^*=-28.433\,1,\quad b^*=0.018\,6.
$$

故

$$
Z(t)=\ 28.433\,1+0.018\,6t.
$$

所以 $$N(t)=e^{-28.4331+0.0186t}.$$

由此可预测出 2000 年人口总数为

$$N(2\,000)\doteq 64.182\times 10^{8}.$$

6.6.3 超定方程组的最小二乘解

在 6.6.2 中我们看到，曲线拟合的理论依据是超定方程组存在最小二乘解，且该最小二乘解与对应的法方程组的解是等价的，以下我们从理论上证明这一结论.

定理 6.2 设 $\boldsymbol{Ax}=\boldsymbol{b}$ 为超定方程组，$\boldsymbol{x}^*$ 为该方程组的最小二乘解的充要条件是 $\boldsymbol{x}^*$ 满足法方程组

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}^*=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}.$$

证 (1)充分性($\Leftarrow$)：

设存在 $\boldsymbol{x}^*$ 满足

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}^*=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}.$$

若 $\tilde{\boldsymbol{x}}=\boldsymbol{x}^*+\boldsymbol{y}(\boldsymbol{y}\neq 0)$，则证明对 $\forall\tilde{\boldsymbol{x}}\neq\boldsymbol{x}^*$ 有

$$\|\boldsymbol{A}\boldsymbol{x}^*-\boldsymbol{b}\|_2^2\leqslant\|\boldsymbol{A}\tilde{\boldsymbol{x}}-\boldsymbol{b}\|_2^2,$$

即 $$\|\boldsymbol{A}\boldsymbol{x}^*-\boldsymbol{b}\|_2^2=\min_{x\neq x^*}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2,$$

亦即 $$\|\boldsymbol{A}\boldsymbol{x}^*-\boldsymbol{b}\|_2=\min_{x\neq x^*}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2,$$

也就是说，$\boldsymbol{x}^*$ 是超定方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 的最小二乘解. 事实上，

$$\begin{aligned}
\|\boldsymbol{A}\tilde{\boldsymbol{x}}-\boldsymbol{b}\|_2^2&=(\boldsymbol{A}\tilde{\boldsymbol{x}}-\boldsymbol{b},\boldsymbol{A}\tilde{\boldsymbol{x}}-\boldsymbol{b})\\
&=(\boldsymbol{A}\tilde{\boldsymbol{x}}-\boldsymbol{b})^{\mathrm{T}}(\boldsymbol{A}\tilde{\boldsymbol{x}}-\boldsymbol{b})\\
&=(\boldsymbol{A}\boldsymbol{x}^*+\boldsymbol{A}\boldsymbol{y}-\boldsymbol{b})^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{x}^*+\boldsymbol{A}\boldsymbol{y}-\boldsymbol{b})\\
&=(\boldsymbol{A}\boldsymbol{x}^*-\boldsymbol{b}+\boldsymbol{A}\boldsymbol{y})^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{x}^*-\boldsymbol{b}+\boldsymbol{A}\boldsymbol{y})\\
&=(\boldsymbol{A}\boldsymbol{x}^*-\boldsymbol{b},\boldsymbol{A}\boldsymbol{x}^*-\boldsymbol{b})+(\boldsymbol{A}\boldsymbol{x}^*-\boldsymbol{b})^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{y})\\
&\quad+(\boldsymbol{A}\boldsymbol{y})^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{x}^*-\boldsymbol{b})+(\boldsymbol{A}\boldsymbol{y})^{\mathrm{T}}\boldsymbol{A}\boldsymbol{y}\\
&=(\boldsymbol{A}\boldsymbol{x}^*-\boldsymbol{b},\boldsymbol{A}\boldsymbol{x}^*-\boldsymbol{b})+2(\boldsymbol{A}\boldsymbol{y})^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{x}^*-\boldsymbol{b})+(\boldsymbol{A}\boldsymbol{y})^{\mathrm{T}}\boldsymbol{A}\boldsymbol{y}\\
&=\|\boldsymbol{A}\boldsymbol{x}^*-\boldsymbol{b}\|_2^2+2\boldsymbol{y}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{x}^*-\boldsymbol{b})+\|\boldsymbol{A}\boldsymbol{y}\|_2^2,
\end{aligned}$$

由假设 $$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}^*=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b},$$

即 $$\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{x}^*-\boldsymbol{b})=\boldsymbol{0}.$$

所以 $$\|\boldsymbol{A}\tilde{\boldsymbol{x}}-\boldsymbol{b}\|_2^2=\|\boldsymbol{A}\boldsymbol{x}^*-\boldsymbol{b}\|_2^2+\|\boldsymbol{A}\boldsymbol{y}\|_2^2\geqslant\|\boldsymbol{A}\boldsymbol{x}^*-\boldsymbol{b}\|_2^2.$$

故 $\boldsymbol{x}^*$ 是 $\boldsymbol{Ax}=\boldsymbol{b}$ 的最小二乘解.

(2)必要性($\Rightarrow$)

设 $\boldsymbol{x}^*$ 是 $\boldsymbol{Ax}=\boldsymbol{b}$ 的最小二乘解，即对 $\forall\boldsymbol{x}\neq\boldsymbol{x}^*$ 有

$$\|\boldsymbol{A}\boldsymbol{x}^*-\boldsymbol{b}\|_2=\min\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2.$$

证明 $\boldsymbol{x}^*$ 满足法方程

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}^*=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}.$$

事实上，$\boldsymbol{Ax}-\boldsymbol{b}$ 的第 i 个分量为

$$\sum_{j=1}^{n} a_{ij} x_j - b_i, \quad i = 1,2,\cdots,m,$$

故
$$\| Ax - b \|_2^2 = (Ax - b, Ax - b) = \sum_{i=1}^{m} (\sum_{j=1}^{n} a_{ij} x_j - b_i)^2.$$

令 $Q = \| Ax - b \|_2^2$，则

$$\frac{\partial Q}{\partial x_k^*} = 2\sum_{i=1}^{m} (\sum_{j=1}^{n} a_{ij} x_j^* - b_i) a_{ik} = 0, \quad k = 1,2,\cdots,n,$$

即
$$A^T A x^* = A^T b.$$

6.6.4 多项式拟合的一般步骤

(1)给定实测数据

$$(x_i, y_i)_{i=1,2,\cdots,m}, \quad x_i \neq x_j (i \neq j), \quad y_i = f(x_i).$$

(2)设立拟合多项式

$$y(x,a) = y(x, a_0, a_1, \cdots, a_n) = a_0 + a_1 x + \cdots + a_n x^n$$
$$= \sum_{j=0}^{n} a_j x^j \quad (n \ll m). \tag{6-30}$$

(3)形成超定方程组，即将数据 $(x_i, y_i), i = 1,2,\cdots,m$ 代入式(6－30)得超定方程组

$$a_0 + a_1 x_i + \cdots + a_n x_i^n = y_i, \quad i = 1,2,\cdots,m,$$

其矩阵形式为 $Aa = y$，其中

$$A = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_m & x_m^2 & \cdots & x_m^n \end{bmatrix},$$

$$a = (a_0, a_1, \cdots, a_n)^T, \quad y = (y_0, y_1, \cdots, y_m)^T.$$

(4)求超定方程组的最小二乘解.

据定理 6.2，a^* 满足

$$\| Aa^* - y \|_2^2 = \min_{a \in \mathbb{R}^n} \| Aa - y \|_2^2$$

的充要条件是 a^* 满足法方程组 $A^T A a^* = A^T y$.

列表形成法方程组(表 6－21 列出的是 $n \leqslant 3$ 的情况).

表 6－21

x_i	y_i	$x_i y_i$	x_i^2	$x_i^2 y_i$	x_i^3	$x_i^3 y_i$	x_i^4	x_i^5	x_i^6
x_1	y_1	$x_1 y_1$	x_1^2	$x_1^2 y_1$	x_1^3	$x_1^3 y_1$	x_1^4	x_1^5	x_1^6
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
x_m	y_m	$x_m y_m$	x_m^2	$x_m^2 y_m$	x_m^3	$x_m^3 y_m$	x_m^4	x_m^5	x_m^6
$\sum x_i$	$\sum y_i$	$\sum x_i y_i$	$\sum x_i^2$	$\sum x_i^2 y_i$	$\sum x_i^3$	$\sum x_i^3 y_i$	$\sum x_i^4$	$\sum x_i^5$	$\sum x_i^6$

例如：当 $n=1, y(x,a)=a_0+a_1x$ 时，

$$A^{T}A=\begin{bmatrix} m & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix},\quad a=(a_0,a_1)^{T},\quad A^{T}y=(\sum y_i, \sum x_i y_i)^{T};$$

当 $n=2, y(x,a)=a_0+a_1x+a_2x^2$ 时，

$$A^{T}A=\begin{bmatrix} m & \sum x_i & \sum x_i^2 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \end{bmatrix},\quad a=(a_0,a_1,a_2)^{T},$$

$$A^{T}y=(\sum y_i, \sum x_i y_i, \sum x_i^2 y_i)^{T};$$

当 $n=3, y(x,a)=a_0+a_1x+a_2x^2+a_3x^3$ 时，

$$A^{T}A=\begin{bmatrix} m & \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & \sum x_i^5 \\ \sum x_i^3 & \sum x_i^4 & \sum x_i^5 & \sum x_i^6 \end{bmatrix},\quad a=(a_0,a_1,a_2,a_3)^{T},$$

$$A^{T}y=(\sum y_i, \sum x_i y_i, \sum x_i^2 y_i, \sum x_i^3 y_i)^{T}.$$

(5)写出拟合多项式

$$y(x,a^*)=a_0^*+a_1^*x+a_2^*x^2+\cdots+a_n^*x^n.$$

6.6.5 曲线化直

多项式拟合比较规范，方法也比较简单，在解决某些实际问题时，可适当对其作些处理. 如有些非线性曲线可以通过适当的变量置换化为线性曲线，从而用直线拟合方法来处理. 对于一个实际的曲线拟合问题，一般先按观测数据描出其散点图，并观察这些散点构成趋势同哪类曲线图形接近，然后选用相接近的曲线拟合方法，通过适当的变量置换转化成直线拟合的问题，按直线拟合解出后再还原成原变量所表示的曲线拟合函数. 例如在 6.6.2 中的例 11(人口预测问题)，所设拟合函数并非多项式，而是指数函数型 $N(t)=e^{a+bt}$，通过两端取对数化成线性函数 $Z(t)=a+bt$，再用直线拟合方法求出 a^*, b^* 后，最后还原成原变量所表示的拟合函数 $N(t)=e^{a^*+b^*t}$.

表 6－22 列举了几类可经适当变换化为线性拟合求解的曲线拟合函数及变换关系.

表 6－22

拟合曲线	变换关系	线性函数
$y=ax^b$	$Y=\lg y, X=\lg x$	$Y=\lg a+bX$
$y=ax^k+c$(k 为常数)	$Y=y, X=x^k$	$Y=aX+c$
$y=ae^{bx}$	$Y=\ln y, X=x$	$Y=bX+\ln a$

续表 6-22

拟合曲线	变换关系	线性函数
$y=a+b\lg x$	$Y=y, X=\lg x$	$Y=a+bX$
$y=\dfrac{x}{ax+b}$	$Y=\dfrac{1}{y}, X=\dfrac{1}{x}$	$Y=a+bX$
$y=\dfrac{e^{a+bx}}{1+e^{a+bx}}$	令 $Y=e^{a+bx}$ $Y=\dfrac{y}{1-y},\quad X=x$	令 $Z=\ln y$, $Z=a+bX$

习题 6

*1. 已知函数 $f(x)=e^{-x^2}$ 的数表如下：

x_i	-0.6	-0.4	-0.2	0	0.2	0.4	0.6
y_i	0.697 676	0.852 114	0.960 789	1	0.960 789	0.852 114	0.697 676

试用二次、三次插值，计算 $x=0.35$ 及 $x=0.55$ 时函数的近似值.

*2. 利用函数 $\sin x$ 在 $x=0,\frac{\pi}{6},\frac{\pi}{4},\frac{\pi}{3}$ 及 $\frac{\pi}{2}$ 处的值，求 $\sin\left(\frac{\pi}{5}\right)$，并估计误差.

*3. 利用以下数表

x_i	0	0.2	0.4	0.6	0.8	1.0
$s(x_i)$	0	0.199 56	0.396 46	0.588 13	0.772 10	0.946 08

计算积分

$$S(x)=\int_0^x \frac{\sin t}{t}\mathrm{d}t$$

当 $S(x)=0.45$ 时，x 的值.

*4. 试用 Newton 插值求经过点 $(-3,-1),(0,2),(3,-2),(6,10)$ 的三次插值公式.

*5. 给出函数 $f(x)$ 的数表如下：

x_i	0.1	0.2	0.3	0.4	0.5
$f(x_i)$	0.700 10	0.401 60	0.108 10	$-0.174\ 40$	$-0.437\ 50$

试利用该数表求方程 $f(x)=0$ 在 0.3 与 0.4 之间的根.

*6. 试构造一个三次 Hermite 插值多项式，使其满足：

$$f(0)=1,\quad f'(0)=0.5,\quad f(1)=2,\quad f'(1)=0.5.$$

*7. 确定一个不高于四次的多项式 $\varphi(x)$，使得

$$\varphi(0)=\varphi'(0)=0,\quad \varphi(1)=\varphi'(1)=\varphi(2)=1.$$

*8. 已知下表中 x_i, y_i 的数据及边界条件 $S'(0.25)=1.0000$, $S'(0.53)$ 0.6868. 试求三次样条函数 $S(x)$.

x_i	0.25	0.30	0.39	0.45	0.53
y_i	0.5000	0.5477	0.6245	0.6708	0.7280

*9. 使电流通过 2Ω 的电阻,用伏特表测量电阻两端的电压,得到如下数据:

I_i	1	2	4	6	8	10
V_i	1.8	3.7	8.2	12.0	15.8	20.2

试用最小二乘法建立 I 与 V 之间的线性经验公式(该公式对于校正测量所用的伏特表有用).

*10. 求下列超定方程组的最小二乘解:

$$\begin{cases}2x+4y=11,\\3x-5y=3,\\x+2y=6,\\2x+y=7.\end{cases}$$

*11. 已知以下数据:

x_i	−1.00	−0.75	−0.50	−0.25	0	0.25	0.50	0.75	1.00
y_i	0.2209	0.3295	0.8826	1.4392	2.0003	2.5645	3.1334	3.7061	4.2836

试用一次、二次、三次多项式按最小二乘原理拟合以上数据,写出法方程组,并求出最小二乘拟合的二次多项式.

*12. 用最小二乘原理求一个形如 $y=a+bx^2$ 的经验公式,使其与下表内数据相拟合.

x_i	19	25	31	38	44
y_i	19.0	32.3	49.0	73.3	97.8

*13. 用最小二乘原理求一个形如 $y=ae^{bx}$ 的经验公式,使其与下表内数据相拟合.

x_i	1	2	3	4
y_i	60	30	20	10

*14. 求函数 $y=\sqrt{x}$ 在区间$[\frac{1}{4},1]$上的最小二乘一次多项式.

15. 设 $l_k(x)=\prod_{\substack{i=0\\i\neq k}}^{n}\frac{x-x_i}{x_k-x_i}$ 为 Lagrange 插值基函数,试证明:

(1) $\sum_{k=0}^{n}l_k(x)=1$;

(2) $\sum_{k=0}^{n} l_k(x)x_k^m = x^m \quad (m = 0,1,\cdots,n)$；

(3) $\sum_{k=0}^{n} l_k(x)(x_k - x)^m = 0 \quad (m = 0,1,\cdots,n)$.

16. 利用 Lagrange 插值多项式与 Newton 插值多项式的最高项系数，证明：

$$f(x_0,x_1,\cdots,x_n) = \sum_{k=0}^{n} \frac{y_k}{\omega'_{n+1}(x_k)}.$$

其中 $$\omega_{n+1}(x) = (x-x_0)(x-x_1)\cdots(x-x_n).$$

17. 设 $f(x) \in C^2[a,b]$ 且 $f(a) = f(b) = 0$，求证：

$$\max_{a\leqslant x\leqslant b}|f(x)| \leqslant \frac{1}{8}(b-a)^2 \max_{a\leqslant x\leqslant b}|f''(x)|.$$

18. 证明 n 阶均差有下列性质：

(1)若 $F(x) = cf(x)$，则

$$F[x_0,x_1,\cdots,x_n] = cf[x_0,x_1,\cdots,x_n];$$

(2)若 $F(x) = f(x) + g(x)$，则

$$F[x_0,x_1,\cdots,x_n] = f[x_0,x_1,\cdots,x_n] + g[x_0,x_1,\cdots,x_n].$$

注　打 * 号的习题在本书附录Ⅱ作了详解.

第 7 章　数值积分

7.1　数值求积公式

7.1.1　关于牛顿－莱布尼兹(Newton－Leibniz)公式

设 $f(x)\in C_{[a,b]}$，则 Newton－Leibniz 公式

$$\int_a^b f(x)\mathrm{d}x = F(x)\Big|_a^b = F(b)-F(a),$$

其中 $F'(x)=f(x)$.

Newton　Leibniz 公式在应用上有很大的局限性：

(1)相当多的被积函数 $f(x)$ 其原函数 $F(x)$ 不能用初等函数表示成有限形式. 例如 $f(x)=\dfrac{\sin x}{x}$，$f(x)=e^{x^2}$，$f(x)=\sin x^2$ 等等；

(2)原函数 $F(x)$ 的表达式相当复杂，不便计算；

(3)被积函数 $f(x)$ 由表格(数据)或图形给出，找不到原函数 $F(x)$ 的表达式.

对于上述情形，Newton－Leibniz 公式或者不能使用，或者不便使用. 因此，在科学研究和工程计算中常采用另一种近似的方法——数值积分.

7.1.2　数值求积公式

由积分中值定理知

$$\int_a^b f(x)\mathrm{d}x = f(\xi)(b-a), \tag{7－1}$$

其中，$\xi\in(a,b)$. 将 $[a,b]$ 细分成 n 个部分，如图 7－1 所示.



图 7－1

令 $f(\xi)\approx\sum\limits_{k=0}^{n}\lambda_k f(x_k)$，其中 λ_k 是 $f(x_k)$ 相应的加权系数. 将 $f(\xi)$ 代入式(7－1)中得

$$\begin{aligned}\int_a^b f(x)\mathrm{d}x &\approx \sum_{k=0}^{n}\lambda_k f(x_k)(b-a)\\ &= \sum_{k=0}^{n}\lambda_k (b-a) f(x_k)\\ &= \sum_{k=0}^{n}A_k f(x_k),\end{aligned} \tag{7－2}$$

其中 $A_k=\lambda_k(b-a)$.

我们称式(7−2)为数值求积公式,A_k 为求积系数. 式(7−2)形式上给出了定积分的近似值,但由于加权系数 λ_k 很难确定,因而求积系数 A_k 也很难确定,因此式(7−2)无实际应用价值.

7.1.3 插值型求积公式

给定实测数据$(x_k,y_k),k=0,1,\cdots,n$,其中

$$x_k\neq x_l(k\neq l),\quad y_k=f(x_k).$$

对于$\forall x\in(a,b)$且 $x\neq x_k$,有

$$f(x)=L_n(x)+R_n(x),$$

其中 $L_n(x)$为 Lagrange 插值多项式,$R_n(x)$为其余项.

对 $f(x)=L_n(x)+R_n(x)$两端积分得

$$\int_a^b f(x)\mathrm{d}x=\int_a^b L_n(x)\mathrm{d}x+\int_a^b R_n(x)\mathrm{d}x.$$

令

$$I=\int_a^b f(x)\mathrm{d}x,$$

$$I_n=\int_a^b L_n(x)\mathrm{d}x=\int_a^b\Big[\sum_{k=0}^n f(x_k)\,l_k(x)\Big]\mathrm{d}x$$

$$=\sum_{k=0}^n\Big[\int_a^b l_k(x)\mathrm{d}x\Big]f(x_k)=\sum_{k=0}^n A_k f(x_k),$$

其中

$$A_k=\int_a^b l_k(x)\mathrm{d}x.$$

令 $I\approx I_n$,即

$$\int_a^b f(x)\mathrm{d}x\approx\sum_{k=0}^n A_k f(x_k). \tag{7-3}$$

我们称式(7−3)为 Lagrange 插值型求积公式.

Lagrange 插值型求积公式具有以下特点:

(1)式中的 $x_k(k=0,1,\cdots,n)$是$[a,b]$上的插值结点;

(2)求积系数 $A_k=\int_a^b l_k(x)\mathrm{d}x$,其中

$$l_k(x)=\prod_{\substack{j=0\\j\neq k}}^n\frac{x-x_j}{x_k-x_j};$$

(3)余项

$$R_{I_n}=I-I_n=\int_a^b[f(x)-L_n(x)]\mathrm{d}x=\int_a^b R_n(x)\mathrm{d}x$$

$$=\int_a^b\frac{f^{(n+1)}(\xi)}{(n+1)!}\omega_{n+1}(x)\mathrm{d}x.$$

综上,插值型求积公式归纳如下:

$$I \approx I_n = \sum_{k=0}^{n} A_k f(x_k),$$

其中
$$A_k = \int_a^b l_k(x)\mathrm{d}x,$$

$$R_{I_n} = \int_a^b \frac{f^{(n+1)}(\xi)}{(n+1)!}\omega_{n+1}(x)\mathrm{d}x.$$

7.1.4 牛顿－柯特斯(Newton－Cotes)公式

为便于上机计算，通常在式(7－3)中取等距离节点，即将积分区间$[a,b]$分成n等份，并令$h=\frac{b-a}{n}$，且让$x_0=a,x_n=b$，则节点为$x_i=a+ih,i=0,1,\cdots,n$.

令$x=a+th,t>0$，且

$$l_k(x) = \prod_{\substack{i=0\\ j\neq k}}^{n} \frac{x-x_j}{x_k-x_j} = \frac{\omega_{n+1}(x)}{(x-x_k)\omega'_{n+1}(x_k)}.$$

因为
$$\begin{aligned}\omega_{n+1}(x) &= (x-a)(x-x_1)\cdots(x-x_n)\\ &= h^{n+1}t(t-1)\cdots(t-n),\end{aligned}$$

$$\omega'_{n+1}(x_k) = (x_k-a)(x_k-x_1)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-b),$$

所以，求积系数

$$A_k = \int_a^b l_k(x)\mathrm{d}x = \frac{(-1)^{n-k}h}{k!(n-k)!}\int_0^n \prod_{\substack{i=0\\ i\neq k}}^{n}(t-i)\mathrm{d}t.$$

将$h=\frac{b-a}{n}$代入上式得

$$A_k = (b-a)C_k^{(n)},$$

其中
$$C_k^{(n)} = \frac{(-1)^{n-k}}{n\,k!\,(n-k)!}\int_0^n \prod_{\substack{i=0\\ i\neq k}}^{n}(t-i)\mathrm{d}t,\quad n=1,2,\cdots\quad k=0,1,2,\cdots,n, \tag{7－4}$$

并称其为 Cotes 系数.

综上，我们得到 Newton－Cotes 公式

$$I = \int_a^b f(x)\mathrm{d}x \approx \sum_{k=0}^{n} A_k f(x_k),$$

其中$A_k=(b-a)C_k^{(n)}$，$C_k^{(n)}$ 按式(7－4)计算.

常用的 Newton－Cotes 公式有以下几种.

(1)梯形公式($n=1,k=0,1$).

节点：　$x_0=a,\quad x_1=b,\quad h=b-a;$

Cotes 系数：　$C_0^{(1)}=C_1^{(1)}=\frac{1}{2};$

求积公式：　$\int_a^b f(x)\mathrm{d}x \approx \frac{b-a}{2}[f(a)+f(b)],$

记为
$$T = \frac{b-a}{2}[f(a)+f(b)]. \tag{7－5}$$

(2)辛普森(Simpson)公式($n=2,k=0,1,2$).

节点：　$x_0=a,\quad x_1=\frac{a+b}{2},\quad x_2=b,\quad h=\frac{b-a}{2};$

Cotes 系数：　$C_0^{(2)}=\frac{1}{6},\quad C_1^{(2)}=\frac{4}{6},\quad C_2^{(2)}=\frac{1}{6};$

求积公式：　$\int_a^b f(x)\mathrm{d}x\approx\frac{b-a}{6}\left[f(a)+4f(\frac{a+b}{2})+f(b)\right],$

记为

$$S=\frac{b-a}{6}\left[f(a)+4f(\frac{a+b}{2})+f(b)\right]. \tag{7-6}$$

(3)3/8Simpson 公式($n=3,k=0,1,2,3$).

节点：　$x_0=a,\quad x_1=a+\frac{b-a}{3},\quad x_2=a+\frac{2}{3}(b-a),\quad x_3=b,\quad h=\frac{b-a}{3};$

Cotes 系数：　$C_0^{(3)}=\frac{1}{8},\quad C_1^{(3)}=\frac{3}{8},\quad C_2^{(3)}=\frac{3}{8},\quad C_3^{(3)}=\frac{1}{8};$

求积公式：

$$\int_a^b f(x)\mathrm{d}x\approx\frac{b-a}{8}\left[f(a)+3f(a+\frac{b-a}{3})+3f(a+\frac{2}{3}(b-a))+f(b)\right],$$

记为

$$S_{3/8}=\frac{b-a}{8}\left[f(a)+3f(a+\frac{b-a}{3})+3f(a+\frac{2}{3}(b-a))+f(b)\right]. \tag{7-7}$$

(4)Cotes 公式($n=4,k=0,1,\cdots,4$).

节点：　$x_0=a,\quad x_1=a+\frac{b-a}{4},\quad x_2=a+\frac{b-a}{2},$

$x_3=a+\frac{3}{4}(b-a),\quad x_4=b,\quad h=\frac{b-a}{4};$

Cotes 系数：　$C_0^{(4)}=C_4^{(4)}=\frac{7}{90},\quad C_1^{(4)}=C_3^{(4)}=\frac{32}{90},\quad C_2^{(4)}=\frac{12}{90};$

求积公式：

$$\int_a^b f(x)\mathrm{d}x\approx\frac{b-a}{90}\left\{7f(a)+7f(b)+32\left[f(a+\frac{b-a}{4})+f(a+\frac{3}{4}(b-a))\right]+12f(a+\frac{b-a}{2})\right\}\overset{\text{记为}}{=\!=}C. \tag{7-8}$$

将上述最终求积公式归纳如下：

$T=\frac{b-a}{2}[f(a)+f(b)],$

$S=\frac{b-a}{6}\left[f(a)+f(b)+4f(\frac{a+b}{2})\right],$

$S_{3/8}=\frac{b-a}{8}\left[f(a)+f(b)+3f(a+\frac{b-a}{3})+3f(a+\frac{2}{3}(b-a))\right],$

$C=\frac{b-a}{90}\left\{7[f(a)+f(b)]+12f(a+\frac{b-a}{2})+32\left[f(a+\frac{b-a}{4})+f(a+\frac{3}{4}(b-a))\right]\right\}.$

7.2 数值求积公式的代数精度

7.2.1 代数精度的概念

定义 7.1(代数精度) 设数值求积公式

$$\int_a^b f(x)\mathrm{d}x \approx \sum_{k=0}^{n} A_k f(x_k).$$

若对次数不超过 m 的多项式 $p_r(x), r \leqslant m$ 均有

$$\int_a^b p_r(x)\mathrm{d}x = \sum_{k=0}^{n} A_k p_r(x_k)$$

成立,则称该式的代数精度至少为 m;

若对次数不超过 m 的多项式 $p_r(x), r \leqslant m$ 均有

$$\int_a^b p_r(x)\mathrm{d}x = \sum_{k=0}^{n} A_k p_r(x_k),$$

而对 $p_r(x), r > m$,上式不一定精确成立,则称该式的代数精度为 m.

例 1 验证梯形公式

$$\int_a^b f(x)\mathrm{d}x \approx \frac{b-a}{2}[f(a)+f(b)]$$

的代数精度为 1.

证 令 $f(x)=1$,代入原式得

$$b-a=b-a;$$

令 $f(x)=x$,代入原式得

$$\frac{1}{2}(b^2-a^2)=\frac{1}{2}(b^2-a^2);$$

令 $f(x)=x^2$,代入原式得

$$\frac{1}{3}(b^3-a^3)\neq\frac{1}{2}(b-a)(a^2+b^2).$$

所以,梯形公式 $I\approx T$ 的代数精度为 1.

同理可以验证:

Simpson 公式 $I\approx S$ 的代数精度为 3;

3/8Simpson 公式 $I\approx S_{3/8}$ 的代数精度为 3;

Cotes 公式 $I\approx C$ 的代数精度为 5.

例 2 设数值求积公式

$$\int_{-h}^{h} f(x)\mathrm{d}x \approx A_{-1}f(-h)+A_0 f(0)+A_1 f(h),$$

试确定求积系数 A_{-1}, A_0, A_1,且使公式的代数精度尽量高.

解 令 $f(x)=1$,使原式精确成立,得

$$A_{-1}+A_0+A_1=2h;$$

令 $f(x)=x$,使原式精确成立,得

$$A_{-1}-A_1=0;$$

令 $f(x)=x^2$，使原式精确成立，得

$$A_{-1}+A_1=\frac{2}{3}h.$$

联立上述三式得线性方程组

$$\begin{cases}A_{-1}+A_0+A_1=2h,\\ A_{-1}-A_1=0,\\ A_{-1}+A_1=\frac{2}{3}h.\end{cases}$$

解得 $$A_{-1}=A_1=\frac{1}{3}h,\quad A_0=\frac{4}{3}h.$$

将其结果代入原式得

$$\int_{-h}^{h}f(x)\mathrm{d}x\approx\frac{h}{3}[f(-h)+4f(0)+f(h)]. \tag{7-9}$$

再令 $f(x)=x^3$，代入式(7－9)，其等式精确成立；$f(x)=x^4$ 代入式(7－9)，其等式不精确成立.所以，式(7－9)的代数精度为 3.

定理 7.1(插值型求积公式的代数精度) 设数值求积公式

$$\int_a^b f(x)\mathrm{d}x\approx\sum_{k=0}^{n}A_k f(x_k),$$

其代数精度至少为 n 的充分必要条件是它为插值型求积公式.

证 首先证明充分性.

设原式是插值型求积公式，则式中的求积系数

$$A_k=\int_a^b l_k(x)\mathrm{d}x,$$

$$\begin{aligned}I_n&=\sum_{k=0}^{n}A_k f(x_k)=\sum_{k=0}^{n}\left[\int_a^b l_k(x)\mathrm{d}x\right]f(x_k)\\&=\int_a^b\left[\sum_{k=0}^{n}l_k(x)f(x_k)\right]\mathrm{d}x\\&=\int_a^b L_n(x)\mathrm{d}x,\end{aligned}$$

余项为

$$R_{I_n}=I-I_n=\int_a^b R_n(x)\mathrm{d}x=\int_a^b\frac{f^{(n+1)}(\xi)}{(n+1)!}\omega_{n+1}(x)\mathrm{d}x.$$

将 $f(x)$ 换成次数不超过 n 的多项式 $p_r(x)(r\leqslant n)$，则

$$R_{I_n}=I-I_n=0,\quad 即\ I\equiv I_n.$$

据定义 7.1，原式的代数精度至少为 n.

其次证必要性.

设原式的代数精度至少为 n，据定义 7.1，对次数不超过 n 的多项式 $p_n(x)(r\leqslant n)$ 原式精确成立，特别地取 Lagrange 插值基函数 $l_k(x)$，原式亦应精确成立，即

$$\int_a^b l_k(x)\mathrm{d}x = \sum_{j=0}^{n} A_j l_k(x_j).$$

因为
$$l_k(x_j)=\begin{cases}1, & i=k,\\ 0, & j\neq k.\end{cases}$$

所以
$$A_k=\int_a^b l_k(x)\mathrm{d}x.$$

故原式 $\int_a^b f(x)\mathrm{d}x \approx \sum_{k=0}^{n} A_k f(x_k)$ 为插值型求积公式.

由定理 7.1,对照前面常用的几个插值型求积公式可知,它们的代数精度至少为 n.

(1)式(7－5):$n=1$,代数精度为 1;

(2)式(7－6):$n=2$,代数精度为 3;

(3)式(7－7):$n=3$,代数精度为 3;

(4)式(7－8):$n=4$,代数精度为 5.

定理 7.2(偶阶 Newton－Cotes 公式的代数精度)　当 n 为偶数时,Newton－Cotes 公式

$$\int_a^b f(x)\mathrm{d}x \approx \sum_{k=0}^{n} A_k f(x_k)$$

(其中 $A_k=(b-a)C_k^{(n)}$,$C_k^{(n)}$ 按式(7－4)计算)的代数精度至少为 $n+1$.

可以证明,当 n 为偶数时,对 $p_r(x)(r\leqslant n+1)$有

$$\int_a^b p_r(x)\mathrm{d}x = \sum_{k=0}^{n} A_k p_r(x_k)$$

成立,即原式的代数精度至少为 $n+1$. 例如:Simpson 公式 $I\approx S$,$n=2$(偶数),其代数精度为 $n+1=3$;Cotes 公式 $I\approx C$,$n=4$(偶数),其代数精度为 $n+1=5$;而 3/8Simpson 公式 $I\approx S_{3/8}$,$n=3$(奇数),其代精度仅为 $n=3$.

7.2.2　插值型求积公式余项估计

如 7.1.3 中所述,插值型求积公式 $I\approx I_n$,其余项

$$R_{I_n}=I-I_n=\int_a^b[f(x)-L_n(x)]\mathrm{d}x$$
$$=\int_a^b R_n(x)\mathrm{d}x$$
$$=\int_a^b \frac{f^{(n+1)}(\xi)}{(n+1)!}\omega_{n+1}(x)\mathrm{d}x.$$

下面就常用的 Newton－Cotes 公式的余项进行估计:

(1)$R_{I_1}=R_T=\int_a^b R_1(x)\mathrm{d}x=\int_a^b \frac{f''(\xi)}{2!}(x-a)(x-b)\mathrm{d}x.$

因为$(x-a)(x-b)\leqslant 0, x\in[a,b]$,据广义积分中值定理

$$R_T=\frac{f''(\eta)}{2!}\int_a^b (x-a)(x-b)\mathrm{d}x=-\frac{f''(\eta)}{12}(b-a)^3.$$

(2)$R_{I_2}=R_S=\int_a^b R_2(x)\mathrm{d}x=\int_a^b \frac{f'''(\xi)}{3!}(x-a)(x-\frac{a+b}{2})(x-b)\mathrm{d}x.$

因为$(x-a)(x-\frac{a+b}{2})(x-b)$在$[a,b]$上不保号，故不能使用广义积分中值定理而改用代数精度的概念. 又因为 $I\approx S$，代数精度为 3，所以

$$\begin{aligned}R_S&=\int_a^b[f(x)-L_3(x)]\mathrm{d}x=\int_a^b R_3(x)\mathrm{d}x\\&=\int_a^b \frac{f^{(4)}(\xi)}{4!}(x-a)(x-\frac{a+b}{2})^2(x-b)\mathrm{d}x\\&=\frac{f^{(4)}(\eta)}{4!}\int_a^b(x-a)(x-\frac{a+b}{2})^2(x-b)\mathrm{d}x\\&=-\frac{f^{(4)}(\eta)}{180}(b-a)(\frac{b-a}{2})^4.\end{aligned}$$

(3)据 $I\approx C$ 的代数精度为 5，可推得

$$R_C=-\frac{2}{945}f^{(6)}(\eta)(b-a)(\frac{b-a}{4})^6.$$

(4)矩形中点公式(中矩公式)$I_0=(b-a)f(\frac{a+b}{2})$，其代数精度为 1，其余项

$$\begin{aligned}R_{I_0}&=\int_a^b R_1(x)\mathrm{d}x=\int_a^b \frac{f''(\xi)}{2}(x-\frac{a+b}{2})^2\mathrm{d}x\\&=\frac{f''(\eta)}{2}\int_a^b(x-\frac{a+b}{2})^2\mathrm{d}x=\frac{f''(\eta)}{24}(b-a)^3.\end{aligned}$$

综上所述，常用的插值型求积公式的余项表达式归纳如下：

$$\begin{aligned}R_T&=-\frac{f''(\eta)}{12}(b-a)^3,\\R_S&=-\frac{f^{(4)}(\eta)}{180}(b-a)(\frac{b-a}{2})^4,\\R_C&=-\frac{2}{945}f^{(6)}(\eta)(b-a)(\frac{b-a}{4})^6,\\R_{I_0}&=\frac{f''(\eta)}{24}(b-a)^3.\end{aligned}$$

7.3 复化求积公式

7.3.1 复化梯形公式

将区间$[a,b]$分为 n 等份，如图 7－2 所示.



图 7－2

设
$$h=x_{k+1}-x_k=\frac{b-a}{n},$$
并在区间$[x_k,x_{k+1}]$上应用梯形公式
$$\int_{x_k}^{x_{k+1}} f(x)\mathrm{d}x \approx \frac{h}{2}[f(x_k)+f(x_{k+1})],\quad k=0,1,\cdots,n-1,$$
所以
$$\int_a^b f(x)\mathrm{d}x = \sum_{k=0}^{n-1}\int_{x_k}^{x_{k+1}} f(x)\mathrm{d}x$$
$$\approx \frac{h}{2}\sum_{k=0}^{n-1}[f(x_k)+f(x_{k+1})]$$
$$\approx \frac{h}{2}[f(a)+f(b)+2\sum_{k=1}^{n-1}f(x_k)].$$
记
$$T_n=\frac{h}{2}[f(a)+f(b)+2\sum_{k=1}^{n-1}f(x_k)], \tag{7-10}$$
并称其为区间$[a,b]$上的复化梯形公式,其余项为
$$R_{T_n}=-\frac{b-a}{12}h^2f''(\eta),\quad h=\frac{b-a}{n}.$$

7.3.2 复化 Simpson 公式

如图 7－3 所示,分别在子区间$[x_k,x_{k+1}]$上应用 Simpson 公式.



图 7－3

$$\int_{x_k}^{x_{k+1}} f(x)\mathrm{d}x \approx \frac{h}{6}[f(x_k)+4f(x_{k+\frac{1}{2}})+f(x_{k+1})],\quad k=0,1,\cdots,n-1,$$
$$\int_a^b f(x)\mathrm{d}x \approx \frac{h}{6}\Big[f(a)+f(b)+4\sum_{k=0}^{n-1}f(x_{k+\frac{1}{2}})+2\sum_{k=1}^{n-1}f(x_k)\Big].$$
记
$$S_n=\frac{h}{6}\Big[f(a)+f(b)+4\sum_{k=0}^{n-1}f(x_{k+\frac{1}{2}})+2\sum_{k=1}^{n-1}f(x_k)\Big], \tag{7-11}$$
$$R_{S_n}=-\frac{b-a}{180}\Big(\frac{h}{2}\Big)^4f^{(4)}(\eta),\quad h=\frac{b-a}{n}.$$

7.3.3 复化 Cotes 公式

如图 7－4 所示,分别在子区间$[x_k,x_{k+1}]$上应用 Cotes 公式.

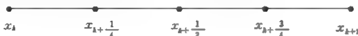


图 7－4

因为
$$\int_{x_k}^{x_{k+1}} f(x)\mathrm{d}x \approx \frac{h}{90}[7f(x_k)+7f(x_{k+1})+32f(x_{k+\frac{1}{4}})+32f(x_{k+\frac{3}{4}})+12f(x_{k+\frac{1}{2}})],$$

所以

$$\int_a^b f(x)\mathrm{d}x \approx \frac{h}{90}\Big[7f(a)+7f(b)+32\sum_{k=0}^{n-1}f(x_{k+\frac{1}{4}})+32\sum_{k=0}^{n-1}f(x_{k+\frac{3}{4}})$$
$$+12\sum_{k=0}^{n-1}f(x_{k+\frac{1}{2}})+14\sum_{k=1}^{n-1}f(x_k)\Big].$$

记

$$C_n=\frac{h}{90}\Big\{7[f(a)+f(b)]+32\Big[\sum_{k=0}^{n-1}f(x_{k+\frac{1}{4}})+\sum_{k=0}^{n-1}f(x_{k+\frac{3}{4}})\Big]+12\sum_{k=0}^{n-1}f(x_{k+\frac{1}{2}})+14\sum_{k=1}^{n-1}f(x_k)\Big\},$$
$$R_{C_n}=-\frac{2(b-a)}{945}\left(\frac{h}{4}\right)^6 f^{(6)}(\eta),\quad h=\frac{b-a}{n}. \tag{7-12}$$

例 3 利用表 7－1 计算

$$I=\int_0^1 \frac{\sin x}{x}\mathrm{d}x.$$

表 7－1

x	0	1/8	1/4	3/8	1/2	5/8	3/4	7/8	1
$f(x)$	1	0.997 397 8	0.989 615 8	0.976 726 7	0.958 851 0	0.936 155 6	0.908 851 6	0.877 192 5	0.841 470 9

解 (1) 将区间[0,1]分为 8 等份，用 $n=8$ 的复化梯形公式(7－10)($h=1/8$)，有

$$T_8=\frac{1}{2}\times\frac{1}{8}\Big\{f(0)+f(1)+2\Big[f(\frac{1}{8})+f(\frac{1}{4})$$
$$+f(\frac{3}{8})+f(\frac{1}{2})+f(\frac{5}{8})+f(\frac{3}{4})+f(\frac{7}{8})\Big]\Big\}$$
$$=0.945\,690\,9.$$

注意计算时应令 $f(x)=\frac{\sin x}{x}, f(0)=\lim\limits_{x\to 0}\frac{\sin x}{x}=1$.

(2) 将区间[0,1]分为 4 等份，用 $n=4$ 的复化 Simpson 公式(7－11)($h=1/4$)，有

$$S_4=\frac{1}{6}\times\frac{1}{4}\Big\{f(0)+f(1)+4\Big[f(\frac{1}{8})+f(\frac{3}{8})+f(\frac{5}{8})+f(\frac{7}{8})\Big]$$
$$+2\Big[f(\frac{1}{4})+f(\frac{1}{2})+f(\frac{3}{4})\Big]\Big\}=0.946\,083\,2.$$

(3) 将区间[0,1]分为 2 等份，用 $n=2$ 的复化 Cotes 公式(7－12)($h=1/2$)，有

$$C_2=\frac{1}{90}\times\frac{1}{2}\Big\{7[f(0)+f(1)]+32\Big[f(\frac{1}{8})+f(\frac{5}{8})\Big]+12\Big[f(\frac{1}{4})+f(\frac{3}{4})\Big]$$
$$+32\Big[f(\frac{3}{8})+f(\frac{7}{8})\Big]+14f(\frac{1}{2})\Big\}=0.946\,083\,1.$$

上述计算结果与精确值 I 比较：

$I=0.946\,083\,1$， 7 位有效数字，

$T_8=0.945\,690\,9$， 2 位有效数字，

$S_4=0.946\,083\,2$， 6 位有效数字，

$C_2=0.946\,083\,1$， 7 位有效数字.

7.3.4 小结

常用 Newton－Cotes 公式、复化公式及其余项小结如下：

$$\begin{cases} I_0 = (b-a)f(a) \quad \text{（左矩公式）} \\ R_{I_0} = \dfrac{f'(\eta)}{2}(b-a)^2 \end{cases}$$

$$\begin{cases} I_0 = (b-a)f(b) \quad \text{（右矩公式）} \\ R_{I_0} = -\dfrac{f'(\eta)}{2}(b-a)^2 \end{cases}$$

$$\begin{cases} I_0 = (b-a)f(\dfrac{a+b}{2}) \quad \text{（中矩公式）} \\ R_{I_0} = \dfrac{f''(\eta)}{24}(b-a)^3 \end{cases}$$

$$\begin{cases} T = \dfrac{b-a}{2}[f(a)+f(b)] \quad \text{（梯形公式）} \\ R_T = -\dfrac{f''(\eta)}{12}(b-a)^3 \end{cases}$$

$$\begin{cases} S = \dfrac{b-a}{6}[f(a)+f(b)+4f(\dfrac{a+b}{2})] \quad \text{（Simpson 公式）} \\ R_S = -\dfrac{1}{90}(\dfrac{b-a}{2})^5 f^{(4)}(\eta) \end{cases}$$

$$\begin{cases} C = \dfrac{b-a}{90}\Big\{7[f(a)+f(b)]+32[f(a+\dfrac{b-a}{4}) \\ \qquad +f(a+\dfrac{3}{4}(b-a))]+12f(a+\dfrac{b-a}{2})\Big\} \quad \text{（Cotes 公式）} \\ R_C = -\dfrac{8}{945}(\dfrac{b-a}{4})^7 f^{(6)}(\eta) \end{cases}$$

$$\begin{cases} T_n = \dfrac{h}{2}[f(a)+f(b)+2\sum_{k=1}^{n-1}f(x_k)] \quad \text{（复化梯形公式）} \\ R_{T_n} = -\dfrac{f''(\eta)}{12}(b-a)h^2, \quad h = \dfrac{b-a}{n} \end{cases}$$

$$\begin{cases} S_n = \dfrac{h}{6}[f(a)+f(b)+4\sum_{k=0}^{n-1}f(x_{k+\frac{1}{2}})+2\sum_{k=1}^{n-1}f(x_k)] \text{（复化 Simpson 公式）} \\ R_{S_n} = -\dfrac{b-a}{180}(\dfrac{h}{2})^4 f^{(4)}(\eta), \quad h = \dfrac{b-a}{n} \end{cases}$$

$$\begin{cases} C_n = \dfrac{h}{90}\{7[f(a)+f(b)]+32\Big[\sum_{k=0}^{n-1}f(x_{k+\frac{1}{4}})+\sum_{k=0}^{n-1}f(x_{k+\frac{3}{4}})\Big] \quad \text{（复化 Cotes 公式）} \\ \qquad +12\sum_{k=0}^{n-1}f(x_{k+\frac{1}{2}})+14\sum_{k=1}^{n-1}f(x_k)\} \\ R_{C_n} = -\dfrac{2(b-a)}{945}\left(\dfrac{h}{4}\right)^6 f^{(6)}(\eta), \quad h = \dfrac{b-a}{n} \end{cases}$$

7.4 龙贝格(Romberg)求积方法

复化求积方法是一种定步长方法，必须预先给出步长 h(或者区间的等份数 n)，而确定 h(或 n) 只有利用余项估计式 $|R_{I_n}|<\varepsilon$.

由于 R_{I_n} 的结构式很复杂，且有不确定的变量 η，故很难或者根本不可能通过上述估计式定出 h(或者 n).

Romberg 等人提出了一种变步长的求积方法. 其思路是：将区间$[a,b]$逐次半分，反复利用复化求积公式进行计算，并用事后估计的方法作为终止法则. 其好处是无须事先定出步长 h，且便于估计误差.

7.4.1 变步长梯形方法(逐次半分法)

1) 梯形值序列及其递推公式

定义 7.2(梯形值序列)　称 $T_1,T_2,T_{2^2},\cdots,T_{2^k},\cdots$ 为梯形值序列，记为

$$\{T_n\}_{n=2^k},\quad k=0,1,\cdots$$

其中，k 为二分次数，n 为等份数.

例如：

T_1 是在区间$[a,b]$上利用梯形公式计算出的积分近似值.

T_2 是 2 等分区间$[a,b]$并利用复化梯形公式计算出的积分近似值.

⋮

$T_n=T_{2^k}$ 是第 k 次二分区间$[a,b]$，共 $2^k=n$ 等份，并用复化梯形公式计算的积分近似值.

$T_{2n}=T_{2^{k+1}}$ 是第 $k+1$ 次二分区间$[a,b]$(如图 7－5)，共 $2^{k+1}=2n$ 等份，并用复化梯形公式计算的积分近似值.

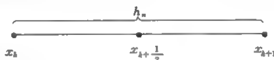


图 7－5

由复化梯形公式有

$$\begin{aligned}T_n&=\frac{h_n}{2}\Big[f(a)+f(b)+2\sum_{k=0}^{n-1}f(x_k)\Big]\\&=\frac{1}{2}\,\frac{b-a}{2^k}\Big[f(a)+f(b)+2\sum_{k=0}^{n-1}f(x_k)\Big],\end{aligned}$$

$$\begin{aligned}T_{2n}&=\frac{1}{2}\cdot\frac{h_n}{2}\Big[f(a)+f(b)+2\sum_{k=1}^{n-1}f(x_k)+2\sum_{k=0}^{n-1}f(x_{k+\frac{1}{2}})\Big]\\&=\frac{1}{2}T_n+\frac{h_n}{2}\sum_{k=0}^{n-1}f(x_{k+\frac{1}{2}})\\&=\frac{1}{2}T_n+\frac{b-a}{2^{k+1}}\sum_{k=0}^{n-1}f(x_{k+\frac{1}{2}}).\end{aligned}$$

由此得递推公式：

$$T_{2n} = \frac{1}{2}T_n + \frac{b-a}{2^{k+1}}\sum_{k=0}^{n-1} f(x_{k+\frac{1}{2}}). \tag{7-13}$$

例如：

$T_1 = \frac{b-a}{2}[f(a) + f(b)]$,

$T_2 = \frac{1}{2}T_1 + \frac{b-a}{2}f(\frac{a+b}{2})$,

$T_4 = \frac{1}{2}T_2 + \frac{b-a}{2^2}\left[f(a + \frac{1}{4}(b-a)) + f(a + \frac{3}{4}(b-a))\right]$,

$\vdots$

由此新增分点如图 7－6 所示.



图 7－6

例 4　用逐次半分法计算 $I = \int_0^1 \frac{\sin x}{x}dx$.

解　按递推公式(7－13)逐次计算如下：

$T_1 = \frac{1}{2}[f(0) + f(1)] = 0.9207355$,

$T_2 = \frac{1}{2}T_1 + \frac{1}{2}f(\frac{1}{2}) = 0.9397933$,

$T_4 = \frac{1}{2}T_2 + \frac{1}{2^2}\left[f(\frac{1}{4}) + f(\frac{3}{4})\right] = 0.9445135$,

$T_8 = \frac{1}{2}T_4 + \frac{1}{2^3}\left[f(\frac{1}{8}) + f(\frac{3}{8}) + f(\frac{5}{8}) + f(\frac{7}{8})\right] = 0.945690$,

$\vdots$

$T_{2^8} = 0.9460831$,

$T_{2^9} = 0.9460831$.

经 9 次二分(共 512 等份)，所得积分近似值 T_{2^9} 具有 7 位有效数字.

2) 算法

Algo. 7.1(逐次半分法)

1°　输入 a, b, ε.

2°　$1 \to n, b-a \to h$.

3°　$h*(f(a)+f(b))/2 \to s$.

4°　$s+\varepsilon+1 \to t$.

5°　$|s-t| \geqslant \varepsilon$?

是，$s \to t$，转 6°；

否，转 8°.

6° $s = \sum_{k=0}^{n-1} f(a + (k + \frac{1}{2})h)$.

7° $t/2 + h * s/2 \to s, h/2 \to h, 2n \to n$,转 5°.

8° 打印 s, stop.

注 Algo. 7.1 中算法的终止法则采用事后估计方法,即

$|T_{2n} - T_n| < \varepsilon$?

是,输出 T_{2n},stop;

否,继续二分直至满足预给的精度为止.

7.4.2 Romberg 方法(逐次半分加速法)

定义 7.3 称

$$\{S_n\}_{n=2^k}, \quad k = 0,1,\cdots$$
$$\{C_n\}_{n=2^k}, \quad k = 0,1,\cdots$$
$$\{R_n\}_{n=2^k}, \quad k = 0,1,\cdots$$

分别叫做 Simpson 积分值序列、Cotes 积分值序列、Romberg 积分值序列. 其中:

S_n 表示将区间$[a,b]$第 k 次二分,共 $2^k = n$ 等份,并用复化梯形公式计算的积分近似值;

C_n 表示将区间$[a,b]$第 k 次二分,共 $2^k = n$ 等份,并用复化 Cotes 公式计算的积分近似值;

R_n 表示将区间$[a,b]$第 k 次二分,共 $2^k = n$ 等份,并用复化 Romberg 公式计算的积分近似值.

(1) 由$\{T_n\}$递推$\{S_n\}$,递推公式为

$$S_n = \frac{4}{4-1}T_{2n} - \frac{1}{4-1}T_n. \tag{7-14}$$

计算步骤如图 7-7 所示

T_1

↓

$T_2 \to S_1$ $\qquad S_1 = \frac{4}{4-1}T_2 - \frac{1}{4-1}T_1$

↓

$T_4 \to S_2$ $\qquad S_2 = \frac{4}{4-1}T_4 - \frac{1}{4-1}T_2$

⋮

↓

$T_{n/2}$

↓

$T_n \to S_{n/2}$ $\qquad S_{n/2} = \frac{4}{4-1}T_n - \frac{1}{4-1}T_{n/2}$

↓

$T_{2n} \to S_n$ $\qquad S_n = \frac{4}{4-1}T_{2n} - \frac{1}{4-1}T_n$

图 7-7

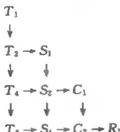
(2) 由$\{S_n\}$递推$\{C_n\}$,递推公式为

$$C_n = \frac{4^2}{4^2-1}S_{2n} - \frac{1}{4^2-1}S_n. \tag{7-15}$$

(3) 由 $\{C_n\}$ 递推 $\{R_n\}$，递推公式为

$$R_n = \frac{4^3}{4^3-1}C_{2n} - \frac{1}{4^3-1}C_n. \tag{7-16}$$

综合上述递推过程图示如下：



此递推过程还可以继续做下去.

7.4.3 Romberg 算法设计

为了表达无穷递推过程，首先必须统一各积分值序列的记号. 记

$T_0^{(i)}(i=0,1,2,\cdots)$ 为梯形积分值序列，

$T_1^{(i)}(i=0,1,2,\cdots)$ 为 Simpson 积分值序列，

$T_2^{(i)}(i=0,1,2,\cdots)$ 为 Cotes 积分值序列，

$T_3^{(i)}(i=0,1,2,\cdots)$ 为 Romberg 积分值序列，

⋮

则 Romberg 算法如下：

(1) 计算
$$T_0^{(0)} = \frac{b-a}{2}[f(a)+f(b)];$$

(2) 计算
$$T_0^{(1)} = \frac{1}{2}T_0^{(0)} + \frac{b-a}{2}f\left(\frac{a+b}{2}\right),$$

一般有
$$T_0^{(i)} = \frac{1}{2}T_0^{(i-1)} + \frac{b-a}{2^i}\sum_{i=0}^{2^{i-1}-1} f(x_{i+\frac{1}{2}});$$

(3) 计算
$$T_1^{(0)} = \frac{4}{4-1}T_0^{(1)} - \frac{1}{4-1}T_0^{(0)},$$

一般有
$$T_1^{(i)} = \frac{1}{4}T_0^{(i+1)} - \frac{1}{4-1}T_0^{(i)};$$

(4) 利用外推公式

$$T_k^{(i)} = \frac{4^k}{4^k-1}T_{k-1}^{(i+1)} - \frac{1}{4^k-1}T_{k-1}^{(i)}$$

逐一向外递推，即 $k=1,2,\cdots,m$，$i=0,1,2,\cdots,m-k$ 直至求得 $T_m^{(0)}$；

(5) 判断：$|T_m^{(0)} - T_{m-1}^{(0)}| < \varepsilon$?

是，令 $I \approx T_m^{(0)}$，

否，重复(2)，(3)，(4) 的计算.

通常将 $T_k^{(i)}$ 排成如下所示的三角形数表(T 表) 再逐次进行计算.

$$\begin{array}{llll}
T_0^{(0)} & & & \\
\downarrow & & & \\
T_0^{(1)} \to & T_1^{(0)} & & \\
\downarrow & \downarrow & & \\
T_0^{(2)} \to & T_1^{(1)} \to & T_2^{(0)} & \\
\downarrow & \downarrow & \downarrow & \\
T_0^{(3)} \to & T_1^{(2)} \to & T_2^{(1)} \to & T_3^{(0)} \\
\vdots & & &
\end{array}$$

可以证明数表 T 沿竖向或对角线方向上的积分值序列均是收敛的，且均收敛于 I.

例 5 用 Romberg 方法计算

$$I = \int_0^1 \frac{4}{1+x^2}\mathrm{d}x.$$

解 将计算结果列于表 7－2.

表 7－2

二分次数 i	等份数 2^i	$T_0^{(i)}$	$T_1^{(i)}$	$T_2^{(i)}$	$T_3^{(i)}$	$T_4^{(i)}$
0	1	3.000 00				
1	2	3.100 00	3.133 33			
2	4	3.131 18	3.141 57	3.142 12		
3	8	3.138 99	3.141 59	3.141 59	3.141 59	
4	16	3.140 94	3.141 59	3.141 59	3.141 59	3.141 59

$$I = \int_0^1 \frac{4}{1+x^2}\mathrm{d}x \approx T_4^{(0)} = 3.141\,59.$$

这一结果与 $\int_0^1 \frac{4}{1+x^2}\mathrm{d}x = \pi$ 相比已有较好的精度.

Algo. 7.2(逐次半分加速法)

1° 输入 a, b, ε.

2° $t_1 \leftarrow \frac{b-a}{2}[f(a)+f(b)]$.

3° $t_0 \leftarrow t_1 + \varepsilon - 1$.

4° $h \leftarrow 2(b-a)$, $m \leftarrow 0$, $n \leftarrow 1$.

5° $|t_{m+1} - t_m| \geqslant \varepsilon$?

是，$h \leftarrow \frac{h}{2}$，$m \leftarrow m+1$，转 6°；

否，转 9°.

6° $p \leftarrow \sum_{i=0}^{n-1} f(a+(i+\frac{1}{2})h)$, $p \leftarrow \frac{t_1}{2} + h * \frac{p}{2}$.

7° $k = 1$ to n, $S \leftarrow \frac{4^k p - t_k}{4^k - 1}$, $t_k \leftarrow p$, $p \leftarrow s$.

8° $t_{m+1} \leftarrow s$, $n \leftarrow 2n$,转 5°.

9° 打印 S.

7.5 Gauss 型求积公式

7.5.1 两点 Gauss 公式

考察两点公式($n = 1$)

$$\int_{-1}^{1} f(x)\mathrm{d}x \approx A_0 f(x_0) + A_1 f(x_1).$$

(1) 若取普通结点 $x_0 = -1, x_1 = 1$,则公式的代数精度为 1,且 $A_0 = 1, A_1 = 1$,即

$$\int_{-1}^{1} f(x)\mathrm{d}x \approx f(-1) + f(1).$$

事实上,上式为区间$[-1,1]$上的梯形公式,故代数精度为 1.

(2) 若取特殊结点 $x_0 = -\frac{1}{\sqrt{3}}, x_1 = \frac{1}{\sqrt{3}}$,则

$$\int_{-1}^{1} f(x)\mathrm{d}x \approx A_0 f(-\frac{1}{\sqrt{3}}) + A_1 f(\frac{1}{\sqrt{3}}). \qquad (*)$$

以下证明该公式的代数精度为 3,且 $A_0 = A_1 = 1$.

事实上,分别令 $f(x) = 1, x$,代入式(*)并令其精确相等,得到二元一次方程组

$$\begin{cases} A_0 + A_1 = 2, \\ -A_0 + A_1 = 0. \end{cases}$$

解此二元一次方程组得 $A_0 = A_1 = 1$.

对于
$$\int_{-1}^{1} f(x)\mathrm{d}x \approx f(-\frac{1}{\sqrt{3}}) + f(\frac{1}{\sqrt{3}}),$$

再分别令 $f(x) = x^2, f(x) = x^3$,该式均精确成立,即

$$\frac{2}{3} = \frac{1}{3} + \frac{1}{3}, \quad 0 = -\frac{1}{3\sqrt{3}} + \frac{1}{3\sqrt{3}}.$$

而令 $f(x) = x^4$ 时,该式不精确成立,即是说

$$\int_{-1}^{1} f(x)\mathrm{d}x \approx f(-\frac{1}{\sqrt{3}}) + f(\frac{1}{\sqrt{3}})$$

具有 3 次代数精度.

该公式具有的代数精度为 3,而不是梯形公式的 1,可从图 7 8 中直观地看出,过两点$\left[(-\frac{1}{\sqrt{3}}), f(-\frac{1}{\sqrt{3}})\right]$,$\left[(\frac{1}{\sqrt{3}}), f(\frac{1}{\sqrt{3}})\right]$的直线 $p_1(x)$ 与 $x = -1, x = 1$ 围成的面积较之区间$[-1,1]$上的梯形面积更接近于 $f(x)$ 在区间$[-1,1]$上的曲边梯形面积.

从以上例子可以看出，在节点数目不变的情况下，求积公式的代数精度是可以提高的.

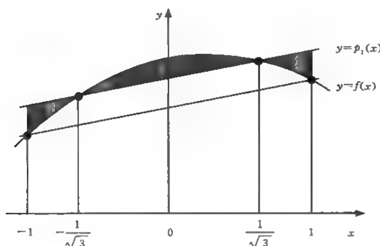


图 7－8

7.5.2 正交多项式

定义 7.4 设$\{p_i(x)\}, i=0,1,\cdots,m$为多项式系，若

$$(p_i(x),p_j(x))=\int_a^b\rho(x)p_i(x)p_j(x)\mathrm{d}x$$
$$=\begin{cases}0, & j\neq i,\\ \text{正数}, & j=i.\end{cases}$$

则称该多项式系为区间$[a,b]$上的关于权函数$\rho(x)$的正交多项式系.

常见的正交多项式系有以下几种：

(1) 切比雪夫(чебышев)多项式系$\{T_k(x)\}$，令$x=\cos\theta, T_n(x)=\cos(n\theta)$，通常用以下递推公式写出：

$$\begin{cases}T_0(x)=1, \quad T_1(x)=x;\\ T_{k+1}(x)=2xT_k(x)-T_{k-1}(x), \quad k\geqslant 2.\end{cases} \tag{7－17}$$

$\{T_k(x)\}$在区间$[-1,1]$上关于权函数$\rho(x)=\frac{1}{\sqrt{1-x^2}}$正交，即

$$(T_n,T_m)=\int_{-1}^1\frac{1}{\sqrt{1-x^2}}T_n(x)T_m(x)\mathrm{d}x=\begin{cases}0, & n\neq m;\\ \frac{\pi}{2}, & n-m\neq 0;\\ \pi, & n=m=0.\end{cases}$$

它的前 7 个多项式为

$$T_0(x)=1,$$

$$T_1(x) = x,$$
$$T_2(x) = 2x^2 - 1,$$
$$T_3(x) = 4x^3 - 3x,$$
$$T_4(x) = 8x^4 - 8x^2 + 1,$$
$$T_5(x) = 16x^5 - 20x^3 + 5x,$$
$$T_6(x) = 32x^6 - 48x^4 + 18x^2 - 1.$$

(2) 勒让德(Legendre) 多项式系$\{p_k(x)\}$,即

$$p_n(x) = \frac{1}{2^n \times n!} \frac{\mathrm{d}^n}{\mathrm{d}x^n}(x^2 - 1)^n.$$

其递推公式如下:

$$\begin{cases} p_0(x) = 1, p_1(x) = x; \\ p_{k+1}(x) = \dfrac{2k+1}{k+1} x p_k(x) - \dfrac{k}{k+1} p_{k-1}(x), \quad k \geqslant 2. \end{cases} \tag{7-18}$$

它的前 7 个多项式为

$$p_0(x) = 1,$$
$$p_1(x) = x,$$
$$p_2(x) = \frac{3x^2 - 1}{2},$$
$$p_3(x) = \frac{5x^3 - 3x}{2},$$
$$p_4(x) = \frac{35x^4 - 30x^2 + 3}{8},$$
$$p_5(x) = \frac{63x^5 - 70x^3 + 15x}{8},$$
$$p_6(x) = \frac{231x^6 - 315x^4 + 105x^2 - 5}{16}.$$

(3) 拉盖尔(Laguerre) 多项式系$\{L_k(x)\}$,即

$$L_n(x) = \mathrm{e}^x \frac{\mathrm{d}^n}{\mathrm{d}x^n}(x^n \mathrm{e}^{-x}).$$

其递推公式如下:

$$\begin{cases} L_0(x) = 1, \quad L_1(x) = 1 - x; \\ L_{k+1}(x) = (2n + 1 - x)L_k(x) - n^2 L_{n-1}(x), \quad k \geqslant 2. \end{cases} \tag{7-19}$$

且它在区间$[0, +\infty)$上关于权函数 $\rho(x) = \mathrm{e} - x$ 正交,即

$$(L_n(x), L_m(x)) = \int_0^{+\infty} \mathrm{e}^{-x} L_n(x) L_m(x) \mathrm{d}x = \begin{cases} 0, & n \neq m; \\ (n!)^2, & n = m. \end{cases}$$

它的前 7 个多项式为

$$L_0(x) = 1,$$
$$L_1(x) = 1 - x,$$

$$L_2(x) = 2 - 4x + x^2,$$
$$L_3(x) = 6 - 18x + 9x^2 - x^3,$$
$$L_4(x) = 24 - 96x + 72x^2 - 16x^3 + x^4,$$
$$L_5(x) = 120 - 600x + 600x^2 - 200x^3 + 25x^4 - x^5,$$
$$L_6(x) = 720 - 4\,320x + 5\,400x^2 - 2\,400x^3 + 450x^4 - 36x^5 + x^6.$$

(4)Hermite 多项式系$\{H_k(x)\}$,即

$$H_k(x) = (-1)^k e^{x^2} \frac{d^k}{dx^k} e^{-x^2}.$$

其递推公式如下:

$$\begin{cases} H_0(x) = 1, \quad H_1(x) = 2x; \\ H_{k+1}(x) = 2xH_k(x) - 2kH_{k-1}(x), \quad k \geqslant 2. \end{cases} \tag{7-20}$$

且它在区间$(-\infty, +\infty)$上关于权函数$\rho(x) = e^{-x^2}$正交,即

$$(H_n, H_m) = \int_{-\infty}^{+\infty} e^{-x^2} H_n(x) H_m(x) dx = \begin{cases} 0, & n \neq m; \\ 2n \times n! \times \sqrt{\pi}, & n = m. \end{cases}$$

它的前 7 个多项式为

$$H_0(x) = 1,$$
$$H_1(x) = 2x,$$
$$H_2(x) = 4x^2 - x,$$
$$H_3(x) = 8x^3 - 12x,$$
$$H_4(x) = 16x^4 - 48x^2 + 12,$$
$$H_5(x) = 32x^5 - 160x^3 + 120x,$$
$$H_6(x) = 64x^6 - 480x^4 + 720x^2 - 120.$$

以上正交多项式的零点全部是单实根,且都分布在它的正交定义区间内,利用这个性质可以构造各种 Gauss 型求积公式.

7.5.3 常用的 Gauss 型求积公式

7.5.3.1 Gauss-Legendre 求积公式

称

$$\int_{-1}^{1} f(x) dx \approx \sum_{k=0}^{n} A_k f(x_k) \tag{7-21}$$

为区间$[-1,1]$上的Gauss-Legendre求积公式.式中x_k为区间$[-1,1]$上的Gauss点(即Legendre 多项式的零点),A_k为与之匹配的求积系数.

可以证明,式(7-21)的代数精度为$2n+1$.

表7-3给出了式(7-21)在$n = 1 \sim 7$时的节点及与之匹配的求积系数.

例如: (1)当$n = 1(k = 0,1)$时,则

$$x_0 = -\frac{1}{\sqrt{3}} = -0.577\,350\,269\,2, \quad x_1 = \frac{1}{\sqrt{3}} = 0.577\,350\,269\,2$$

$$A_0 = A_1 = 1.$$

(2) 当 $n = 2(k = 0,1,2)$ 时，则

$$A_0 = 0.555\,555\,555\,6,\quad x_0 = -0.774\,596\,669\,2;$$
$$A_1 = 0.888\,888\,888\,9,\quad x_1 = 0;$$
$$A_2 = 0.555\,555\,555\,6,\quad x_2 = 0.774\,596\,669\,2.$$

表 7－3

n	$x_i(i = 0,1,\cdots,n)$	$A_i(i = 0,1,\cdots,n)$
1	±0.577 350 269 2	1
2	±0.774 596 669 2	0.555 555 555 56
	0	0.888 888 888 9
3	±0.861 136 311 6	0.347 854 845 1
	±0.339 981 043 6	0.652 145 154 9
4	±0.932 469 514 2	0.236 926 885 1
	±0.538 469 310 1	0.478 628 670 5
	0	0.568 888 888 9
5	±0.932 469 514 2	0.171 324 492 4
	±0.661 209 386 5	0.360 761 573 0
	±0.238 619 186 1	0.467 913 934 6
6	±0.949 107 912 3	0.129 484 966 2
	±0.741 531 185 6	0.279 705 391 5
	±0.405 845 154 0	0.381 830 050 3
	0	0.417 959 183 7
7	±0.960 289 856 5	0.101 228 536 3
	±0.796 666 477 4	0.222 381 034 5
	±0.525 532 409 9	[illegible].313 706 645 9
	±0.183 434 642 5	0.362 687 834

区间$[-1,1]$上的Gauss－Legendre公式经过变换后可导出一般区间$[a,b]$上的Gauss－Legendre 公式.

设

$$x = \frac{b-a}{2}t + \frac{a+b}{2},$$

$$\int_a^b f(x)\mathrm{d}x = \frac{b-a}{2}\int_{-1}^{1} f\left(\frac{b-a}{2}t + \frac{a+b}{2}\right)\mathrm{d}t$$
$$\approx \frac{b-a}{2}\sum_{k=0}^{n} A_k f\left(\frac{b-a}{2}t_k + \frac{a+b}{2}\right). \qquad (7-22)$$

式中，$t_k(k = 0,1,\cdots,n)$ 是区间$[-1,1]$上的 Gauss 点；A_k 是与之匹配的求积系数(t_k，A_k 仍按表 7－3 查找). 我们称式(7－22) 为区间$[a,b]$上的 Gauss－Legendre 公式.

例 6　利用两点 Gauss－Legendre 公式计算

$$I = \int_0^1 \frac{\sin x}{x}\mathrm{d}x.$$

解　因为 $f(x)=\frac{\sin x}{x}$ 为偶函数，则

$$\begin{aligned} I&=\int_0^1\frac{\sin x}{x}dx=\frac{1}{2}\int_{-1}^1\frac{\sin x}{x}dx\\ &\approx\frac{1}{2}\left[\frac{\sin(-0.577\,735\,026\,92)}{-0.577\,735\,026\,92}+\frac{\sin(0.577\,735\,026\,92)}{0.577\,735\,026\,92}\right]\\ &=0.945\,363. \end{aligned}$$

另外也可直接利用式(7－22)计算，即

$$\begin{aligned} I&=\int_0^1\frac{\sin x}{x}dx\\ &\approx\frac{1}{2}\left\{\frac{\sin\left[\frac{1}{2}\times(-0.577\,735\,026\,92)+\frac{1}{2}\right]}{\frac{1}{2}\times(-0.577\,735\,026\,92)+\frac{1}{2}}+\frac{\sin(\frac{1}{2}\times0.577\,735\,026\,92+\frac{1}{2})}{\frac{1}{2}\times0.577\,735\,026\,92+\frac{1}{2}}\right\}\\ &=0.945\,363. \end{aligned}$$

例 7　分别用 3 点及 6 点 Gauss－Legendre 公式计算

$$I=\int_1^3\frac{dy}{y}.$$

解　(1) 用 3 点 Gauss－Legendre 公式，有

$$I=\int_1^3\frac{dy}{y}=\int_{-1}^1\frac{dt}{t+2}=\frac{A_0}{t_0+2}+\frac{A_1}{t_1+2}+\frac{A_2}{t_2+2}$$

成立，据表 7－3 得

$$\begin{cases}t_0=-0.774\,596\,669\,2,\\ t_1=0,\\ t_2=0.774\,596\,669\,2;\end{cases}$$

$$\begin{cases}A_0=0.555\,555\,555\,6,\\ A_1=0.888\,888\,888\,9,\\ A_2=0.555\,555\,555\,6.\end{cases}$$

并将其代入得

$$I=\int_1^3\frac{dy}{y}\approx1.089\,40.$$

(2) 用 6 点 Gauss－Legendre 公式计算

$$I=\int_1^3\frac{dy}{y}\approx1.098\,62.$$

积分值 I 精确至 10 位小数的结果是 1.098 612 289.

7.5.3.2　Gauss－Laguerre *求积公式*

称

$$\int_0^{+\infty}e^{-x}f(x)dx\approx\sum_{k=0}^{n}A_kf(x_k)\qquad(7-23)$$

为区间$[0,+\infty)$上带权函数 $\rho(x)=e^{-x}$ 的 Gauss－Laguerre 求积公式. 式中，x_k 为区间$[0,+\infty)$上的 Gauss－Laguerre 结点(即 Laguerre 多项式的零点)；A_k 为与之匹配的求积系数.

表 7－4 给出了公式(7－23) 在 $n=1\sim4$ 时的节点及与之匹配的求积系数.

表 7－4

n	$x_i(i=0,1,\cdots,n)$	$A_i(i=0,1,\cdots,n)$
1	0.585 786 437 6	0.853 553 390 6
	3.414 213 562 4	0.146 446 609 4
2	0.415 774 556 8	0.711 093 009 9
	2.294 280 360 3	0.278 517 733 6
	6.289 945 082 9	0.010 389 256 3
3	0.322 547 689 6	0.603 154 104 3
	1.745 761 101 2	0.357 418 692 4
	4.536 620 296 9	0.038 887 908 5
	9.395 070 912 3	0.000 539 294 7
4	0.263 560 319 7	0.521 755 610 6
	1.413 403 059 1	0.398 666 811 1
	3.596 425 771 0	0.075 942 449 7
	7.085 810 005 9	0.003 611 758 7
	12.640 800 844 3	0.000 023 370 6

例 8 用 Gauss－Laguerre 求积公式计算

$$I=\int_0^{+\infty}e^{-x}\sin x\mathrm{d}x.$$

解 据式(7－23) 可得

$$I\approx\sum_{k=0}^{n}A_k\sin x_k.$$

表 7－5 列出了 $n=1,2,3,4$ 的计算结果.

表 7－5

n	1	2	3	4
节点	2 点	3 点	4 点	5 点
I_n	0.432 459	0.496 030	0.540 488	0.498 093

注 已知 $\int_0^{+\infty}e^{-x}\sin x\mathrm{d}x$ 的精确值为 0.5.

7.5.3.3 Gauss－Hermite 求积公式

称

$$\int_{-\infty}^{+\infty}e^{-x^2}f(x)\mathrm{d}x\approx\sum_{k=0}^{n}A_k f(x_k) \quad (7-24)$$

为区间$(-\infty,+\infty)$上带权函数 $\rho(x)=e^{-x^2}$ 的 Gauss－Hermite 求积公式. 式中，x_k 为区间$(-\infty,+\infty)$上的 Gauss－Hermite 节点(即是 Hermite 多项式的零点)；A_k 为与之匹配的求积系数.

表 7－6 给出了公式(7－24) 在 $n=1\sim7$ 的节点及与之匹配的求积系数.

表 7－6

n	$x_i(i=0,1,\cdots,n)$	$A_i(i=0,1,\cdots,n)$
1	±0.707 106 781 2	0.886 226 925 5
2	±1.224 744 871 4	0.295 408 975 2
	0	1.181 635 900 6
3	±1.650 680 123 9	0.081 312 835 5
	±0.524 647 623 3	0.084 904 090 0
4	±2.020 182 870 5	0.019 953 242 1
	±0.958 572 464 6	0.393 619 323 2
	0	0.945 308 720 5
5	±2.350 604 973 7	0.004 530 009 9
	±1.335 849 074 0	0.157 067 320 3
	±0.436 077 411 9	0.724 629 509 52
6	±2.651 961 356 8	0.000 971 781 2
	±1.673 551 628 8	0.054 515 582 8
	±0.816 287 882 9	0.425 607 252 6
	0	0.810 264 617 6
7	±2.930 637 420 3	0.000 199 604 1
	±1.981 656 756 7	0.017 077 983 0
	±1.157 193 712 4	0.207 802 325 8
	±0.381 186 990 2	0.661 147 012 6

例 9 用 Gauss－Hermite 求积公式计算

$$I=\int_{+\infty}^{-\infty}e^{-x^2}\sin^2 x\mathrm{d}x.$$

解 据式(7－24)可得

$$I\approx\sum_{k=0}^{n}A_k\sin^2 x_k.$$

表 7－7 列出了 $n=1,3,4,7$ 的计算结果.

表 7－7

n	1	3	4	7
节点	2 点	4 点	5 点	8 点
I_n	0.748 026	0.565 510	0.560 255	0.560 202

注 已知$\int_{-\infty}^{+\infty}e^{-x^2}\sin^2 x\mathrm{d}x=\frac{\pi(1-e^{-1})}{2}\approx 0.560\ 202\ 28.$

习题 7

*1. 分别用复化梯形公式、复化 Simpson 公式计算下列积分，并保留 9 位有效数字.

(1) $\int_0^1 \frac{x}{4+x^2}\mathrm{d}x \quad (n=8)$；　　(2) $\int_0^1 \sqrt{x}\,\mathrm{d}x \quad (n=10)$；

(3) $\int_0^1 \mathrm{e}^{-x^2}\mathrm{d}x \quad (n=10)$；　　(4) $\int_0^{\frac{\pi}{2}} \sqrt{4-\sin^2 x}\,\mathrm{d}x \quad (n=6)$；

(5) $\int_0^{\frac{\pi}{2}} \frac{\sin x}{x}\mathrm{d}x \quad (n=8)$.

* 2. 试用 Romberg 求积公式计算积分 $\frac{2}{\sqrt{\pi}}\int_0^1 \mathrm{e}^{-x^2}\mathrm{d}x$，要求误差不超过 0.5×10^{-5}.

* 3. 试分别用梯形公式、Simpson 公式与 Romberg 算法计算积分

$$\int_0^1 0.5\sqrt{x}\,\mathrm{d}x.$$

* 4. 试确定系数 A_1, A_2, A_3，并使求积公式

$$\int_{-1}^1 f(x)\mathrm{d}x = A_1 f(-1) + A_2 f\left(-\frac{1}{3}\right) + A_3 f\left(\frac{1}{3}\right)$$

具有 2 次代数精度.

* 5. 利用 Gauss - Legendre 求积公式计算积分 $\int_{-4}^4 \frac{1}{1+x^2}\mathrm{d}x$. 分别取 $n=1,2,3$，并与真实值 $2\arctan 4 \approx 2.6516354$ 比较.

6. 确定下列求积公式中的待定参数，使其代数精度尽量高.

(1) $\int_{-h}^h f(x)\mathrm{d}x \approx A_{-1}f(-h) + A_0 f(0) + A_1 f(h)$，

(2) $\int_{-2h}^{2h} f(x)\mathrm{d}x \approx A_{-1}f(-h) + A_0 f(0) + A_1 f(h)$，

(3) $\int_{-1}^1 f(x)\mathrm{d}x \approx \frac{1}{3}[f(-1) + 2f(x_1) + 3f(x_2)]$，

(4) $\int_0^h f(x)\mathrm{d}x \approx \frac{h}{2}[f(0) + f(h)] + ah^2[f'(0) - f'(h)]$.

7. 推导下列三种矩形求积公式.

(1) $\int_a^b f(x)\mathrm{d}x = (b-a)f(a) + \frac{f'(\eta)}{2}(b-a)^2$，

(2) $\int_a^b f(x)\mathrm{d}x = (b-a)f(b) + \frac{f'(\eta)}{2}(b-a)^2$，

(3) $\int_a^b f(x)\mathrm{d}x = (b-a)f\left(\frac{a+b}{2}\right) + \frac{f''(\eta)}{24}(b-a)^3$.

8. 利用 Gauss 型求积公式，计算下列积分的值，分别取 $n=2,3,4$.

(1) $\int_0^{+\infty} \mathrm{e}^{-x}\sqrt{x}\,\mathrm{d}x$；　　(2) $\int_0^{+\infty} \mathrm{e}^{-10x}\sin x\,\mathrm{d}x$；

(3) $\int_{-\infty}^{+\infty} \mathrm{e}^{-x^2}\sqrt{1+x^2}\,\mathrm{d}x$；　　(4) $\int_0^{+\infty} \mathrm{e}x^{-x^2}\cos x\,\mathrm{d}x$.

注　打 * 号的习题在本书附录 Ⅱ 作了详解.

第 8 章　常微分方程初值问题的数值解法

8.1　初值问题与数值解

8.1.1　一阶常微分方程的初值问题

称
$$\begin{cases} y' = f(x,y), & a \leqslant x \leqslant b, \\ y(a) = y_0. \end{cases} \tag{8-1}$$
为一阶常微分方程的初值问题. 其中 $f(x,y)$ 为已知函数, $y(a) = y_0$ 为初始条件(或定解条件).

设 $f(x,y)$ 在区域
$$D_0 = \{(x,y) \mid a \leqslant x \leqslant b, \mid y \mid < \infty\}$$
内连续且对变量 y 满足李普希兹(Lipschitz)条件，即存在常数 L，对 D_0 内的任何两点(x, y_1) 和(x,y_2)，有以下不等式成立：
$$\mid f(x,y_1) - f(x,y_2) \mid \leqslant L \mid y_1 - y_2 \mid,$$
则初值问题(8－1)的解 $y(x)$ 存在且惟一.

从几何上看，初值问题(8－1)的解是通过点(x_0, y_0)的一条曲线，如图 8－1 所示.

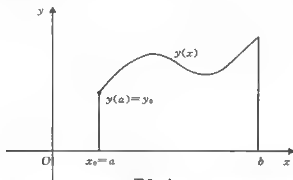


图 8－1

初值问题(8－1)所涉及的一阶微分方程可分为一阶线性微分方程和一阶非线性微分方程两类.

8.1.1.1　一阶线性微分方程

一阶线性微分方程的一般形式为
$$y' = P(x)y + Q(x),$$
其中右端函数是关于 y 的线性函数，$P(x)$ 称为变系数，$Q(x)$ 称为非齐次项.

例 1　马尔萨斯(Multhus)人口方程为

$$
\begin{cases}
y' = ry(x), \\
y(x_0) = y_0.
\end{cases}
\tag{8-2}
$$

式中，x 表示时间，$y(x)$ 表示随时间增长的人口数量，常数 r 表示人口自然增长率.

方程(8－2)称为一阶线性常系数($P(x) = r$)齐次($Q(x) = 0$)微分方程. 其初值问题的解为 $y(x) = y_0 e^{r(x-x_0)}$. 这个解与第 6 章 6.2 中用作人口预测的拟合公式 $N(t) = e^{at+b}$ 是一致的.

Multhus 人口方程(8－2)用作人口预测有很大的局限性，因为在环境等因素的制约下，人口不可能按指数函数规律无限制地增长.

8.1.1.2　一阶非线性微分方程

一阶非线性微分方程的一般形式为

$$y' = f(x,y).$$

其中右端函数 $f(x,y)$ 是关于 y 的非线性函数.

例 2　罗杰斯蒂克(Logistic)人口方程为

$$
\begin{cases}
y' = ry\left(1 - \dfrac{y}{k}\right) = ry - \dfrac{r}{k}y^2, \\
y(x_0) = y_0.
\end{cases}
\tag{8-3}
$$

Logistic 人口方程的右端函数含有 y 的非线性项，式中 r 为人口自然增长率，k 为环境容纳量. 比如在 20 世纪 80 年代初期，我国人口自然增长率 $r = 1.4\%$，而人口专家认同我国人口环境容纳量为 36 亿. Logistic 人口模型由于考虑到了环境对人口增长的限制，因而可用于中长期人口预测.

类似的一阶非线性微分方程还有伯努利(Beroulli)方程

$$y' = P(x)y + Q(x)y^{\alpha}.$$

式中，$P(x)$，$Q(x)$ 为变系数，α 为任意实数.

黎卡堤(Riccati)方程

$$y' = P(x)y + Q(x)y^2 + R(x).$$

该方程较之 Beroulli 方程多了一个非齐次项.

上述一阶非线性微分方程，在一定条件下，通过变量置换可化成一阶线性微分方程求解. 但在一般情形下，一阶非线性微分方程寻求解析解是很困难的，有的甚至不可能求出它的解析解. 即使是一阶线性微分方程，其解析解往往无法用初等函数表达因而不实用. 例如，容易求出初值问题

$$
\begin{cases}
y' = 1 - y\cos t, \quad 0 \leqslant t \leqslant T, \\
y(0) = 0
\end{cases}
$$

的解为

$$y(t) = e^{-\sin t}\int_0^t e^{\sin x}\mathrm{d}x.$$

但是，对给定的 t，要计算 $y(t)$ 的值还是要用数值积分的方法.

鉴于上述情况，研究一阶常微分方程的初值问题，即式(8－1)的数值解法就十分必要了.

8.1.2 数值解与数值解法

8.1.2.1 数值解

在初值问题(8 - 1) 中,设其精确解为 $y(x)$,$x \in [a,b]$,它是满足该初值问题的解析解. 若将区间$[a,b]$插入分点,即

$$x_n = x_0 + nh,\quad n = 0,1,2,\cdots,N,$$

则称 $y(x_n)$,$n = 0,1,\cdots,N$ 为初值问题(8 - 1) 的离散解;称

$$\begin{cases} y_n \approx y(x_n), & n = 1,2,\cdots,N, \\ y_0 = \eta(\text{给定}) \end{cases} \tag{8-4}$$

为初值问题(8 - 1) 的数值解.

显然,$y(x_n)$ 是 $y(x)$ 在各分点处的离散值;而 y_n 则是 $y(x_n)$ 满足一定精度的近似值. 精确解 $y(x)$,离散解 $y(x_n)$ 及数值解 y_n 三者之间的关系如图(8 - 2) 所示.

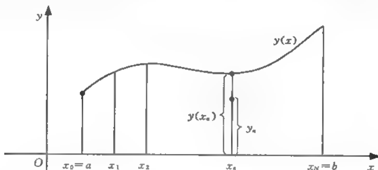


图 8 - 2

8.1.2.2 数值解法

求解初值问题(8 - 1) 的数值解,通常是通过求解差分方程来实现的. 而差分方程的一般形式为

$$F(x_n, y_n, y_{n+1}, \cdots, y_{n+k}, h) = 0,\quad n = 0,1,\cdots,N-k. \tag{8-5}$$

其中,k 是一正整数,N 是区间等份数,函数 F 与初值问题(8 - 1) 中的微分方程的右端函数 $f(x,y)$ 有关. 方程(8 - 5) 称为关于 $y_0, y_1, \cdots, y_N$ 的差分方程. 数值解法的实质是用关于 $y_0, y_1, \cdots, y_N$ 的差分方程(8 - 5) 近似代替初值问题(8 - 1) 中的微分方程,并从 $y_0, y_1, \cdots, y_{k-1}$ 出发,从式(8 - 5) 中依次逐个解出 $y_k, y_{k+1}, \cdots, y_N$,从而得到初值问题(8 - 1) 的数值解式(8 - 4).

上述求解过程的具体做法如下:

令 $n = 0$,由 $F(x_0, y_0, y_1, \cdots, y_{k-1}, y_k, h) = 0$ 解出 y_k(以 $y_0, y_1, \cdots, y_{k-1}$ 为出发值);

令 $n = 1$,由 $F(x_1, y_1, y_2, \cdots, y_k, y_{k+1}, h) = 0$ 解出 y_{k+1}(以 $y_1, y_2, \cdots, y_k$ 为出发值);

⋮

令 $n = N - k$,由 $F(x_{N-k}, y_{N-k}, y_{N-k+1}, \cdots, y_{N-1}, y_N, h) = 0$ 解出 y_N(以 $y_{N-k}, y_{N-k+1}, \cdots, y_{N-1}$ 为出发值).

若 $k=1$,式(8-5)为

$$F(x_n,y_n,y_{n+1},h)=0,\quad n=0,1,\cdots,N-1. \tag{8-6}$$

通常称式(8-6)为单步公式(一个出发值),它是本章研究的重点;若 $k\geqslant 2$,则式(8-5)统称为多步公式(多个出发值).

若单步公式(8-6)能表示为

$$y_{n+1}=\varphi(x_n,y_n,h),\quad n=0,1,\cdots,N-1, \tag{8-7}$$

或
$$y_{n+1}=y_n+h\phi(x_n,y_n,h),\quad n=0,1,\cdots,N-1.$$

则称式(8-7)为显式公式,相应地称式(8-6)为隐式公式.

8.2 欧拉(Euler)公式与梯形公式

8.2.1 Euler 公式(显式与隐式)

将区间$[a,b]$等分为 N 个小区间,取步长 $h=\dfrac{b-a}{N}$,节点 $x_n=x_0+nh,n=0,1,2,\cdots,N$.

设 $y(x)$ 为初值问题(8-1)的精确解,在子区间$[x_n,x_{n+1}]$上对微分方程 $y'=f(x,y)$ 两端积分,有

$$\int_{x_n}^{x_{n+1}}y'(x)\mathrm{d}x=\int_{x_n}^{x_{n+1}}f(x,y(x))\mathrm{d}x,$$

即
$$y(x_{n+1})-y(x_n)=\int_{x_n}^{x_{n+1}}f(x,y(x))\mathrm{d}x. \tag{8-8}$$

记 $g(x)=f(x,y(x))$,并对式(8-8)的右端运用左矩形公式(见第7章7.3.4中的左矩公式)

$$\int_{x_n}^{x_{n+1}}g(x)\mathrm{d}x=hg(x_n)+\frac{g'(\xi)}{2}h^2,\quad \xi\in(x_n,x_{n+1}),$$

则
$$y(x_{n+1})-y(x_n)=hf(x_n,y(x_n))+\frac{g'(\xi)}{2}h^2.$$

令 $y(x_n)=y_n,y(x_{n+1})\approx y_{n+1}$,舍去$\dfrac{g'(\xi)}{2}h^2$,得到

$$y_{n+1}=y_n+hf(x_n,y_n),\quad n=0,1,\cdots,N-1. \tag{8-9}$$

称式(8-9)为显式 Euler 公式,简称 Euler 公式.

Euler 公式还可由泰勒(Taylor)展开式得到.假定 $y(x)$ 二阶连续可导,将 $y(x_{n+1})$ 在 x_n 点展开

$$y(x_{n+1})=y(x_n+h)=y(x_n)+y'(x_n)h+\frac{y''(\xi)}{2!}h^2.$$

令 $y(x_n)=y_n$,舍去 h^2 项,并注意到 $y'(x_n)=f(x_n,y(x_n))=f(x_n,y_n)$,可得

$$y_{n+1}=y_n+hf(x_n,y_n),\quad n=0,1,\cdots,N-1.$$

若对式(8-8)的右端用右矩形公式代替积分,得到

$$y(x_{n+1})-y(x_n)=hf(x_{n+1},y(x_{n+1}))-\frac{g'(\xi)}{2}h^2.$$

同理可得

$$y_{n+1} = y_n + hf(x_{n+1}, y_{n+1}), \quad n = 0,1,\cdots,N-1. \tag{8-10}$$

称式(8－10)为隐式Euler公式.由于公式的两端都含有 y_{n+1},在一般情形下求解 y_{n+1} 很困难,需要解非线性方程.

8.2.2 两步 Euler 公式(Euler 中点公式)

设 $y(x)$ 为初值问题(8－1)的精确解,在子区间$[x_{n-1}, x_{n+1}]$上对微分方程 $y' = f(x,y)$ 两端积分,有

$$\int_{x_{n-1}}^{x_{n+1}} y'(x)\mathrm{d}x = \int_{x_{n-1}}^{x_{n+1}} f(x,y(x))\mathrm{d}x,$$

$$y(x_{n+1}) - y(x_{n-1}) = \int_{x_{n-1}}^{x_{n+1}} f(x,y(x))\mathrm{d}x. \tag{8-11}$$

仍记 $g(x) = f(x,y(x))$,并对式(8－11)的右端运用中矩公式(见第7章7.3.4中的中矩公式)

$$\int_{x_{n-1}}^{x_{n+1}} g(x)\mathrm{d}x = 2hg(x_n) + \frac{g''(\xi)}{24}h^3,$$

所以

$$y(x_{n+1}) - y(x_{n-1}) = 2hf(x_n, y(x_n)) + \frac{g''(\xi)}{24}h^3.$$

令 $y(x_{n-1}) = y_{n-1}, y(x_n) = y_n$,舍去$\frac{g''(\xi)}{24}h^3$,得到

$$y_{n+1} = y_{n-1} + 2hf(x_n, y_n), \quad n = 1,2,\cdots,N-1. \tag{8-12}$$

称式(8－12)为两步Euler公式(或Euler中点公式).显然,利用该式计算 y_{n+1} 时需要两个初值 y_{n-1} 和 y_n.

在利用上述公式计算 y_{n+1} 时,若只需一个初值(出发值),则称该式为单步 Euler 公式(或单步法);若需两个以上的初值(出发值),则称该公式为多步 Euler 公式(多步法).式(8－9)及(8－10)为单步公式,(8－12)为两步公式.

8.2.3 梯形公式

将显示 Euler 公式(8－9)与隐式 Euler 公式(8－10)相加即得

$$y_{n+1} = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1})], \quad n = 0,1,\cdots,N-1. \tag{8-13}$$

称式(8－13)为梯形公式.

梯形公式也可由式(8－8)直接推出,即对式(8－8)的右端用梯形求积公式代替积分得到

$$y(x_{n+1}) - y(x_n) = \frac{h}{2}[f(x_n, y(x_n)) + f(x_{n+1}, y(x_{n+1}))] - \frac{g''(\xi)}{6}h^3.$$

舍去 $-\frac{g''(\xi)}{6}h^3$ 项,得到

$$y_{n+1} = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1})].$$

这个公式的左、右两端均含有 y_{n+1}，因此它也是一个隐式公式.

将上述公式归纳如下：

(1) 显式、单步 Euler 公式：

$$y_{n+1} = y_n + hf(x_n, y_n), \quad n = 0,1,\cdots,N-1;$$

(2) 隐式、单步 Euler 公式：

$$y_{n+1} = y_n + hf(x_{n+1}, y_{n+1}), \quad n = 0,1,\cdots,N-1;$$

(3) 显式、两步 Euler 公式：

$$y_{n+1} = y_{n-1} + 2hf(x_n, y_n), \quad n = 1,2,\cdots,N-1;$$

(4) 隐式、单步梯形公式：

$$y_{n+1} = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1})], \quad n = 0,1,\cdots,N-1.$$

8.3 Euler 方法及其改进方法

8.3.1 Euler 方法

8.3.1.1 Euler 公式的计算表格

将式(8－9)改写成如下形式：

$$\begin{cases} y_{n+1} = y_n + hK, \\ K = f(x_n, y_n), \quad n = 0,1,\cdots,N-1. \end{cases} \tag{8－14}$$

其表格计算方案见表 8－1.

表 8－1

x_n	y_n	$K = f(x_n, y_n)$
x_0	$y_0 = \eta$	$K = f(x_0, y_0)$
$x_1 = x_0 + h$	$y_1 = y_0 + hK$	$K = f(x_1, y_1)$
$x_2 = x_1 + h$	$y_2 = y_1 + hK$	$K = f(x_2, y_2)$
⋮	⋮	⋮
$x_N = x_{N-1} + h$	$y_N = y_{N-1} + hK$	

例 3 用 Euler 公式(8－14)解初值问题

$$\begin{cases} y' = y - \dfrac{2x}{y}, \quad x \in [0,1]; \\ y(0) = 1. \end{cases}$$

分析 该方程为 Beroulli 方程，即

$$y' = P(x)y + Q(x)y^{\alpha},$$

其中

$$P(x) = 1, \quad Q(x) = -2x, \quad \alpha = -1.$$

令 $Z = y^{1-\alpha} = y^2$，将其线性化，得一阶线性微分方程

$$Z' - 2Z - 4x.$$

其解析解为 $Z = 1 + 2x$，即

$$y = \sqrt{1+2x}.$$

以下用数值方法求解,并与其解析解的结果进行比较.

解 将区间[0,1]分成 10 等份,并令 $h = 0.1$,则 Euler 公式为

$$\begin{cases} y_{n+1} = y_n + 0.1K, \\ K = y_n - \dfrac{2x_n}{y_n}. \end{cases}$$

按表 8－1 的格式列表计算,见表 8－2.

表 8－2

x_n	y_n	$K = y_n - \dfrac{2x_n}{y_n}$
$x_0 = 0$	$y_0 = 1$	$K = 1 - 2 \times \dfrac{0}{1} = 1$
$x_1 = 0.1$	$y_1 = 1 + 0.1K = 1.1$	$K = 1.1 - 2 \times \dfrac{0.1}{1.1} = 0.918\,18$
$x_2 = 0.2$	$y_2 = 1.1 + 0.1K = 1.191\,818$	$K = 1.191\,818 - \dfrac{2 \times 0.2}{1.191\,818} = 0.856\,196$
⋮	⋮	⋮
$x_{10} = 1.0$	$y_{10} = 1.784\,771$	

例 3 的全部计算结果 $y_n (n = 1,2,\cdots,10)$ 示于表 8－3,该表分别列出了 Euler 公式的计算结果和精确解 $y = \sqrt{1+2x}$ 的计算结果.

表 8－3

x_i	Euler 方法	精确值
0.1	1.100 000	1.095 445
0.2	1.191 818	1.183 216
0.3	1.277 438	1.264 911
0.4	1.358 213	1.341 641
0.5	1.435 133	1.414 214
0.6	1.508 966	1.483 240
0.8	1.649 783	1.612 452
0.9	1.717 779	1.673 320
1.0	1.784 771	1.732 051

例 4 分别用显式 Euler 公式、隐式 Euler 公式和两步 Euler 公式解初值问题

$$\begin{cases} y' = -y, \quad x \in [0,1], \\ y(0) = 1. \end{cases}$$

解 (1) 显示 Euler 公式($h=0.1$).

$$\begin{cases} y_{n+1}=y_n+0.1K, \\ K=-y_n. \end{cases}$$

此法按表 8－1 计算,其结果见表 8－5 第 3 列.

(2) 隐式 Euler 公式($h=0.1$).

由 $y_{n+1}=y_n-0.1y_{n+1}$ 解出 y_{n+1} 得

$$\begin{cases} y_{n+1}=\dfrac{y_n}{1+0.1}=\dfrac{y_n}{1.1}, \quad n=0,1,\cdots,9, \\ y_0=1. \end{cases}$$

此法不必列表,直接按以上公式计算,其计算结果见表 8－5 第 4 列.

(3) 两步 Euler 公式($h=0.1$).

该公式需要两个初值(出发值),故先由 Euler 公式计算 y_1,然后运用两步公式

$$\begin{cases} y_{n+1}=y_{n-1}-2hy_n, \quad n=1,2,\cdots,9, \\ y_0=1. \end{cases}$$

先将以上公式改写成

$$\begin{cases} y_{n+1}=y_{n-1}+2hK, \\ K=-y_n. \end{cases}$$

其表格计算格式如表 8－4 所示.

表 8－4

x_{n-1}	y_{n-1}	y_n	$K=-y_n$
x_0	y_0	y_1	$K=-y_1$
$x_1=x_0+h$	y_1	$y_2=y_0+2hK$	$K=-y_2$
$x_2=x_1+h$	y_2	$y_3=y_1+2hK$	$K=-y_3$
⋮	⋮	⋮	⋮
$x_N=x_{N-1}+h$	y_{N-1}	$y_N=y_{N-2}+2hK$	

计算结果见表 8－5 第 5 列.

表 8－5

x_n	$y(x_n)=e^{-x_n}$	显式 Euler 法	隐式 Euler 法	两步 Euler 法
0.1	0.904 837	0.900 000	0.909 091	0.900 000
0.2	0.818 731	0.810 000	0.826 446	0.820 000
0.3	0.740 818	0.729 000	0.751 315	0.726 000
0.4	0.670 320	0.656 000	0.683 013	0.672 800
0.5	0.606 531	0.590 490	0.620 921	0.601 440
0.6	0.548 812	0.531 441	0.564 474	0.552 512
0.7	0.496 585	0.478 297	0.513 158	0.490 938

续表 8－5

0.8	0.449 329	0.430 467	0.466 507	0.454 324
0.9	0.406 570	0.387 420	0.424 098	0.400 073
1.0	0.367 879	0.348 678	0.385 543	0.374 310

注 本例中的初值问题的解析解(精确解)为 $y = e^{-x}$.

8.3.1.2 Euler 方法算法设计

Algo. 8.1(动态存储方式)

1° 送初值 x_0, y_0, h, N;打印 x_0, y_0.

2° $1 \to i$.

3° $f(x_0, y_0) \to K$.

4° $x_0 + h \to x_1$, $y_0 + Kh \to y_1$.

5° 打印 x_1, y_1.

6° $i + 1 \to i$.

7° $i > N$?

是,转 8°;

否,$x_1 \to x_0, y_1 \to y_0$,转 3°.

8° Stop.

Algo. 8.2(数组存储方式)

1° 送初值 x_0, y_0, h, N.

2° $0 \to i$.

3° $f(x(i), y(i)) \to K$.

4° $x(i) + h \to x(i+1)$, $y(i) + hK \to y(i+1)$.

5° $i + 1 \to i$.

6° $i \geqslant N$?

是,转 7°;

否,转 3°.

7° 打印 $x(i), y(i) (i = 0, 1, \cdots, N)$.

8° Stop.

注 Algo. 8.2 需引入数组:

	$x(N)$	$y(N)$
初值	$x(0)$	$y(0)$
计算值	$x(1), \cdots,$	$x(N)$
	$y(1), \cdots,$	$y(N)$

8.3.2 改进的 Euler 方法

用 Euler 公式作预报,用梯形公式作校正,从而构成预报－校正公式

$$\begin{cases}\bar{y}_{n+1} = y_n + hf(x_n, y_n),\\ y_{n+1} = y_n + \dfrac{h}{2}[f(x_n, y_n) + f(x_{n+1}, \bar{y}_{n+1})].\end{cases}$$

将预报－校正公式改写成便于表格计算的形式

$$\begin{cases}y_{n+1} = y_n + \dfrac{h}{2}(K_1 + K_2),\\ K_1 = f(x_n, y_n),\\ K_2 = f(x_n + h, y_n + hK_1), \quad n = 0,1,\cdots,N-1.\end{cases} \tag{8－15}$$

多次校正的预报－校正公式

$$\begin{cases}y_{n+1}^{(0)} = y_n + hf(x_n, y_n),\\ y_{n+1}^{(k+1)} = y_n + \dfrac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1}^{(k)})].\end{cases}$$

式中，$n = 0,1,\cdots,N-1, k = 0,1,\cdots$

8.3.2.1 Euler 方法(一次校正)的计算表格

公式(8－15)的计算表格如表 8－6 所示.

表 8－6

x_n	y_n	$K_i(i=1,2)$	$K^* = \frac{1}{2}(K_1 + K_2)$
x_0 $x_0 + h$	y_0 $y_0 + hK_1$	$K_1 = f(x_0, y_0)$ $K_2 = f(x_0 + h, y_0 + hK_1)$	K^*
$x_1 = x_0 + h$ $x_1 + h$	$y_1 = y_0 + hK^*$ $y_1 + hK_1$	$K_1 = f(x_1, y_1)$ $K_2 = f(x_1 + h, y_1 + hK_1)$	K^*
$x_2 = x_1 + h$	$y_2 = y_1 + hK^*$		
⋮	⋮	⋮	
$x_N = x_{N-1} + h$	$y_n = y_{N-1} + hK^*$		

例 5 用改进的 Euler 公式(8－15)解初值问题

$$\begin{cases}y' = y - \dfrac{2x}{y}, \quad x \in [0,1],\\ y(0) = 1.\end{cases}$$

解 首先按式(8－15)写出求解公式($h = 0.1$)

$$\begin{cases}y_{n+1} = y_n + \dfrac{0.1}{2}(K_1 + K_2),\\ K_1 = y_n - \dfrac{2x_n}{y_n},\\ K_2 = (y_n + 0.1K_1) - \dfrac{2(x_n + 0.1)}{y_n + 0.1K_1}.\end{cases}$$

按表(8　6)计算，其计算结果列于表 8－7，并与 Euler 方法和初值问题的精确解

$y=\sqrt{1+2x}$ 的计算结果进行比较.

表 8 - 7

x_n	Euler 方法	改进 Euler 方法	精确值
0.1	1.100 000	1.095 909	1.095 445
0.2	1.191 818	1.184 097	1.183 216
0.3	1.277 438	1.266 201	1.264 911
0.4	1.358 213	1.343 360	1.341 641
0.5	1.435 133	1.416 402	1.414 214
0.6	1.508 966	1.485 956	1.483 240
0.7	1.580 338	1.552 515	1.549 193
0.8	1.649 783	1.616 478	1.612 453
0.9	1.717 779	1.678 167	1.673 320
1.0	1.784 771	1.737 686	1.732 051

8.3.2.2 改进 Euler 方法的算法设计

Algo. 1.3(动态存储)

1° 送初值 x_0, y_0, h, N.

2° $1 \to i$.

3° $f(x_0, y_0) \to K_1$, $f(x_0+h, y_0+h) \to K_2$.

4° $x_0 + h \to x_1$, $y_0 + \frac{h}{2}(K_1+K_2) \to y_1$.

5° 打印 x_1, y_1.

6° $i+1 \to i$.

7° $i > N$?

是,转 8°;

否,$x_1 \to x_0, y_1 \to y_0$,转 3°.

8° Stop.

公式(8 - 15) 也可以写成如下形式:

$$\begin{cases} y_p = y_n + hf(x_n, y_n), \\ y_c = y_n + hf(x_{n+1}, y_p), \\ y_{n+1} = \frac{1}{2}(y_p + y_c). \end{cases}$$

其中,y_p 为预报值(prediction),y_c 为校正值(correction),y_{n+1} 为第 n 次递推值.

8.4 单步方法的截断误差与阶

定义 8.1(局部截断误差) 设单步公式(8 - 6)

$$F(x_n, y_n, y_{n+1}, h) = 0, \quad n = 0,1,\cdots,N-1,$$

或(8 − 7)

$$y_{n+1} = \varphi(x_n, y_n, h), \quad n = 0,1,\cdots,N-1.$$

若 $y(x)$ 是初值问题(8 − 1)的精确解，则称 $y(x_{n+1}) - y_{n+1}$ 为上述单步公式的截断误差；若 $y(x_n) \neq y_n$，则称 $y(x_{n+1}) - y_{n+1}$ 为整体截断误差；若 $y(x_n) = y_n$，则称 $y(x_{n+1}) - y_{n+1}$ 为局部截断误差.

定义 8.2(单步公式的阶) 设单步公式(8 − 6)或(8 − 7)的局部截断误差为

$$y(x_{n+1}) - y_{n+1} = \frac{y^{(p+1)}(\xi_n)}{(p+1)!}h^{p+1} \approx \frac{y^{(p+1)}(x_n)}{(p+1)!}h^{p+1} = o(h^{p+1}),$$

则称该单步公式具有 p 阶精度，且其局部截断误差的首项为

$$\frac{y^{(p+1)}(x_n)}{(p+1)!}h^{p+1}.$$

例 6 试确定 Euler 公式、隐式 Euler 公式、两步 Euler 公式的阶.

解 (1)Euler 公式

$$y_{n+1} = y_n + h f(x_n, y_n).$$

当 $y_n = y(x_n)$ 有

$$y_{n+1} = y(x_n) + h f(x_n, y(x_n)) = y(x_n) + hy'(x_n).$$

由 Taylor 展开式

$$\begin{aligned} y(x_{n+1}) &= y(x_n + h) \\ &= y(x_n) + hy'(x_n) + \frac{h^2}{2}y''(\xi_n), \quad \xi_n \in (x_n, x_{n+1}). \end{aligned}$$

所以

$$y(x_{n+1}) - y_{n+1} = \frac{1}{2}y''(\xi_n)h^2 \approx \frac{1}{2}y''(x_n)h^2 = o(h^2).$$

由定义 8.2，该 Euler 公式具有一阶精度，且其截断误差的首项为 $\frac{1}{2}y''(x_n)h^2$.

(2) 两步 Euler 公式

$$y_{n+1} = y_{n-1} + 2h f(x_n, y_n).$$

设 $y_{n-1} = y(x_{n-1}), y_n = y(x_n)$，有 $y_{n+1} = y(x_{n-1}) + 2hy'(x_n)$.

将

$$y(x_{n-1}) = y(x_n) - hy'(x_n) + \frac{h^2}{2}y''(x_n) - \frac{h^3}{3!}y'''(\xi_n)$$

代入上式得

$$y_{n+1} = y(x_n) + hy'(x_n) + \frac{h_2}{2}y''(x_n) - \frac{h^3}{3!}y'''(\xi_n),$$

而

$$y(x_{n+1}) = y(x_n) + hy'(x_n) + \frac{h^2}{2}y''(x_n) + \frac{h^3}{3!}y'''(\xi_n),$$

所以

$$y(x_{n+1}) - y_{n+1} = \frac{h^3}{3}y'''(\xi_n) \approx \frac{h^3}{3}y'''(x_n) = o(h^3).$$

由定义 8.2，该两步 Euler 公式具有二阶精度，且其截断误差首项为$\frac{h^3}{3}y'''(x_n)$.

(3) 隐式 Euler 公式

$$y_{n+1} = y_n + hf(x_{n+1}, y_{n+1}).$$

设 $$y_n = y(x_n).$$

将 $f(x_{n+1}, y_{n+1})$ 作如下变换：

$$\begin{aligned} f(x_{n+1}, y_{n+1}) &= f(x_{n+1}, y(x_{n+1}) + (y_{n+1} - y(x_{n+1}))) \\ &= f(x_{n+1}, y(x_{n+1})) + f_y(x_{n+1}, \eta)(y_{n+1} - y(x_{n+1})), \end{aligned}$$

其中，η 介于 y_{n+1} 与 $y(x_{n+1})$ 之间. 所以

$$y_{n+1} = y(x_n) + h[f(x_{n+1}, y(x_{n+1})) + f_y(x_{n+1}, \eta)(y_{n+1} - y(x_{n+1}))]. \tag{8-16}$$

而 $$\begin{aligned} f(x_{n+1}, y(x_{n+1})) &= y'(x_{n+1}) \\ &= y'(x_n) + hy''(x_n) + \frac{h^2}{2}y'''(x_n) + \cdots \end{aligned}$$

将其代入式(8－16)得

$$y_{n+1} = hf_y(x_{n+1}, \eta)(y_{n+1} + y(x_{n+1})) + y(x_n) + hy'(x_n) + h^2 y''(x_n) + \frac{h^3}{2}y'''(x_n) + \cdots \tag{8-17}$$

而 $$y(x_{n+1}) = y(x_n) + hy'(x_n) + \frac{h^2}{2}y''(x_n) + \frac{h^3}{3!}y'''(x_n) + \cdots \tag{8-18}$$

用式(8－18)减去式(8－17)得

$$\begin{aligned} y(x_{n+1}) - y_{n+1} &= -hf_y(x_{n+1}, \eta)(y_{n+1} - y(x_{n-1})) + (\frac{1}{2} - 1)h^2 y''(x_n) \\ &\quad + (\frac{1}{3!} - \frac{1}{2})h^3 y'''(x_n) + \cdots \\ &= -hf_y(x_{n+1}, \eta)(y_{n+1} - y(x_{n+1})) - \frac{1}{2}h^2 y''(x_n) + \cdots \end{aligned}$$

整理上式得

$$[1 - hf_y(x_{n+1}, \eta)](y(x_{n+1}) - y_{n+1}) = -\frac{1}{2}h^2 y''(x_n) + \cdots \tag{8-19}$$

据 $$\frac{1}{1-x} = 1 + x + x^2 + \cdots$$

得 $$\frac{1}{1 - hf_y(x_{n+1}, \eta)} = 1 + hf_y(x_{n+1}, \eta) + \cdots$$

并将其代入式(8－19)得

$$y(x_{n+1}) - y_{n+1} = -\frac{1}{2}h^2 y''(x_n) - \frac{1}{2}h^3 y''(x_n) f_y(x_{n+1}, \eta) + \cdots$$

所以 $$y(x_{n+1}) - y_{n+1} \approx -\frac{1}{2}h^2 y''(x_n) = o(h^2).$$

故该隐式 Euler 公式具有一阶精度，且其截断误差的首项为 $\frac{1}{2}h^2 y''(x_n)$.

(4) 梯形公式

$$y_{n+1} = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1})].$$

用类似于(3) 中的方法可证得

$$y(x_{n+1}) - y_{n+1} \approx -\frac{h^3}{12}y'''(x_n) = o(h^3),$$

即梯形公式具有二阶精度，且其截断误差的首项为 $-\frac{h^3}{12}y'''(x_n)$.

归纳上述结论如下：

Euler 公式：$y(x_{n+1}) - y_{n+1} \approx \frac{y''(x_n)}{2}h^2 = o(h^2)$，一阶精度；

隐式 Euler 公式：$y(x_{n+1}) - y_{n+1} \approx -\frac{y''(x_n)}{2}h^2 = o(h^2)$，一阶精度；

两步 Euler 公式：$y(x_{n+1}) - y_{n+1} \approx \frac{y'''(x_n)}{3}h^3 = o(h^3)$，二阶精度；

梯形公式：$y(x_{n+1}) - y_{n+1} \approx -\frac{y'''(x_n)}{12}h^3 = o(h^3)$，二阶精度.

8.5 龙格－库塔(Runge－Kutta)方法

8.5.1 Runge－Kutta(简称 R－K)方法的基本思路

对于初值问题(8－1)，在区间$[x_n, x_{n+1}]$上应用微分中值定理(如图 8－3)：

$$y(x_{n+1}) - y(x_n) = y'(\xi_n)(x_{n+1} - x_n).$$

其中，$\xi_n = x_n + \theta(x_{n+1} - x_n)$，$\theta \in (0,1)$，$y'(\xi_n)$ 称为 $y(x)$ 在区间$[x_n, x_{n+1}]$上的平均变化率.

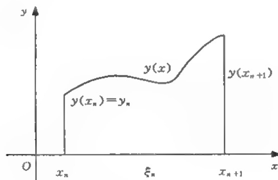


图 8－3

令 $h = x_{n+1} - x_n$，则有

$$\begin{aligned} y(x_{n+1}) &= y(x_n) + hy'(x_n + \theta h) \\ &= y(x_n) + hf(x_n + \theta h, y(x_n + \theta h)). \end{aligned} \tag{8-20}$$

在式(8－20)中，若取 $y'(x_n + \theta h) \approx y'(x_n)$，则有

$$y_{n+1} = y_n + hy'(x_n) = y_n + hf(x_n, y_n).$$

此即显式 Euler 公式；若取

$$y'(x_n + \theta h) \approx \frac{1}{2}[y'(x_n) + y'(x_{n+1})],$$

则有
$$y_{n+1} = y_n + \frac{h}{2}[y'(x_n) + y'(x_{n+1})]$$
$$= y_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1})],$$
此即梯形公式.

将显示 Euler 公式与梯形公式结合形成改进的 Euler 公式:
$$\begin{cases} y_{n+1} = y_n + \frac{h}{2}(K_1 + K_2), \\ K_1 = f(x_n, y_n), \\ K_2 = f(x_n + h, y_n + hK_1). \end{cases}$$

由上述显式 Euler 公式与梯形公式的推导得到启示,若在区间$[x_n, x_{n+1}]$内多预测几个点的斜率值,如 $K_i(i = 1,2,\cdots,N)$,即令
$$y'(x_n + \theta h) \approx \sum_{i=1}^{N} \lambda_i K_i,$$
则有可能构造出高精度的计算公式.

8.5.2 二阶 R－K 公式

二阶 R－K 公式即是将改进的 Euler 公式进一步推广,推导出具有二阶精度的各种近似公式.

下面考虑区间$[x_n, x_{n+1}]$上的两点:x_n 与 $x_{n+p}(x_{n+p} = x_n + ph, 0 \leqslant p \leqslant 1)$,并在式(8－20)中令
$$y'(x_n + \theta h) = \lambda_1 K_1 + \lambda_2 K_2.$$
其中
$$K_1 = y'(x_n) = f(x_n, y_n),$$
$$K_2 = y'(x_{n+p}) = f(x_{n+p}, y(x_{n+p})) = f(x_{n+p}, y_n + phK_1).$$
于是我们得到二阶 R－K 公式的一般形式
$$\begin{cases} y_{n+1} = y_n + h(\lambda_1 K_1 + \lambda_2 K_2), \\ K_1 = f(x_n, y_n), \\ K_2 = f(x_{n+p}, y_n + phK_1). \end{cases} \tag{8－21}$$

可以证明,欲使式(8－21)具有二阶精度,则式中的三个待定系数 λ_1, λ_2, p 应满足不定方程
$$\begin{cases} \lambda_1 + \lambda_2 = 1, \\ \lambda_2 p = \frac{1}{2}. \end{cases} \tag{8－22}$$

事实上,在式(8－21)中
$$K_2 = f(x_n + ph, y_n + phK_1)$$
$$= f(x_n, y_n) + \frac{\partial f(x_n, y_n)}{\partial x} ph + \frac{\partial f(x_n, y_n)}{\partial y} phK_1 + \cdots$$
$$= f(x_n, y_n) + ph\left(\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} \cdot f\right)_{(x_n, y_n)} + \cdots$$

$$= f_n + ph(f_x' + f_y' \cdot f)_{(x_n,y_n)} + \cdots$$

将 $K_1 = f(x_n, y_n) = f_n$ 及 K_2 代入式(8－21)中得

$$y_{n+1} = y_n + (\lambda_1 + \lambda_2)hf_n + \lambda_2 ph^2(f_x' + f_y' \cdot f)_{(x_n,y_n)} + \cdots$$

另一方面,按 Taylor 展式

$$y(x_{n+1}) = y(x_n) + hy'(x_n) + \frac{h^2}{2}y''(x_n) + \cdots$$

$$= y_n + hf_n + \frac{h^2}{2}(f_x' + f_y' \cdot f)_{(x_n,y_n)} + \cdots$$

由此得局部截断误差

$$R(x_{n+1}) = y(x_{n+1}) - y_{n+1}$$

$$= (1 - \lambda_1 - \lambda_2)hf_n + (\frac{1}{2} - \lambda_2 p)h^2(f_x' + f_y' \cdot f)_{(x_n,y_n)} + o(h^3).$$

据定义 8.2,欲使式(8－21)具有二阶精度,则有

$$1 - \lambda_1 - \lambda_2 = 0, \quad \frac{1}{2} - \lambda_2 p = 0,$$

即满足式(8－22)

$$\begin{cases} \lambda_1 + \lambda_2 = 1, \\ \lambda_2 p = \frac{1}{2}; \end{cases}$$

反之亦真.

我们称满足式(8－22)条件的一族公式(8－21)为二阶 R－K 公式.下面列出它的几种常用形式.

1) 改进的 Euler 公式

在式(8－21)中,取 $p = 1$,按式(8－22)解得 $\lambda_1 = \lambda_2 = \frac{1}{2}$,于是有

$$\begin{cases} y_{n+1} = y_n + \frac{h}{2}(K_1 + K_2), \\ K_1 = f(x_n, y_n), \\ K_2 = f(x_n + h, y_n + hK_1). \end{cases}$$

2) 变形的 Euler 公式(中点公式)

取 $p = \frac{1}{2}$,按式(8－22)解得 $\lambda_1 = 0, \lambda_2 = 1$,则有

$$\begin{cases} y_{n+1} = y_n + hK_2, \\ K_1 = f(x_n, y_n), \\ K_2 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_1). \end{cases} \qquad (8-23)$$

3) Heun(海伦)公式

取 $p = \frac{2}{3}$,按式(8－22)解得 $\lambda_1 = \frac{1}{4}, \lambda_2 = \frac{3}{4}$,则有

$$
\begin{cases}
y_{n+1} = y_n + \frac{h}{4}(K_1 + 3K_2), \\
K_1 = f(x_n, y_n), \\
K_2 = f(x_n + \frac{2}{3}h, y_n + \frac{2}{3}hK_1).
\end{cases} \tag{8-24}
$$

8.5.3 四阶 R－K 公式

用类似的方法可以确定三阶和四阶 R－K 公式的参数，从而构造三阶和四阶的 R－K 方法. 下面介绍两个四阶 R－K 公式.

1) 经典公式

$$
\begin{cases}
y_{n+1} = y_n + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4), \\
K_1 = f(x_n, y_n), \\
K_2 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_1), \\
K_3 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_2), \\
K_4 = f(x_n + h, y_n + hK_3).
\end{cases} \tag{8-25}
$$

式中，$n = 0,1,\cdots,N-1$.

2) 吉尔(Gill) 公式

$$
\begin{cases}
y_{n+1} = y_n + \frac{h}{6}[K_1 + (2-\sqrt{2})K_2 + (2+\sqrt{2})K_3 + K_4], \\
K_1 = f(x_n, y_n), \\
K_2 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_1), \\
K_3 = f(x_n + \frac{h}{2}, y_n + \frac{\sqrt{2}-1}{2}hK_1 + (1-\frac{\sqrt{2}}{2})hK_2), \\
K_4 = f(x_n + h, y_n - \frac{\sqrt{2}}{2}hK_2 + (1+\frac{\sqrt{2}}{2})hK_3),
\end{cases} \tag{8-26}
$$

式中，$n = 0,1,\cdots,N-1$.

例 7 用经典公式(8－25)解初值问题

$$
\begin{cases}
y' = y - \frac{2x}{y}, \quad x \in [0,1]; \\
y_0 = 1, \quad h = 0.2.
\end{cases}
$$

解 由式(8－25)得

$$\begin{cases} y_{n+1} = y_n + \frac{0.2}{6}(K_1 + 2K_2 + 2K_3 + K_4), \\ K_1 = y_n - \frac{2x_n}{y_n}, \\ K_2 = (y_n + 0.1K_1) - 2\frac{x_n + 0.1}{y_n + 0.1K_1}, \\ K_3 = (y_n + 0.1K_2) - 2\frac{x_n + 0.1}{y_n + 0.1K_2}, \\ K_4 = (y_n + 0.2K_3) - 2\frac{x_n + 0.2}{y_n + 0.2K_3}. \end{cases}$$

式中,$n = 0,1,2,3,4,5$.其计算方案如表 8－8 所示,全部计算结果见表 8－9.

表 8－8

x_n	y_n	$K_i(i = 1,2,3,4)$	K^*
$x_0 = 0$	$y_0 = 1$	$K_1 = 1$	
$x_0 + 0.1 = 0.1$	$y_0 + 0.1 \times K_1 = 1.1$	$K_2 = 1.1 - 2 \times \frac{0.1}{1.1} = 0.918\,182$	
$x_0 + 0.1 = 0.1$	$y_0 + 0.1 \times K_2 = 1.091\,818$	$K_3 = 1.091\,818 - 2 \times \frac{0.1}{1.091\,818} = 0.908\,637$	
$x_0 + 0.2 = 0.2$	$y_0 + 0.2 \times K_3 = 1.181\,727$	$K_4 = 1.181\,727 - 2 \times \frac{0.2}{1.181\,727} = 0.843\,240$	$K^* = \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4) = 0.916\,146$
$x_1 = x_0 + 0.2 = 0.2$	$y_1 = y_0 + 0.2 \times K^* = 1.183\,229$		
⋮	⋮		
$x_5 = 1.0$	$y_5 = 1.732\,142$		

表 8－9

x_n	y_n	$y(x_n)$
$x_0 = 0$	$y_0 = 1$	$y(x_0) = 1$
$x_1 = 0.2$	$y_1 = 1.183\,229$	$y(x_1) = 1.183\,216$
$x_2 = 0.4$	$y_2 = 1.341\,667$	$y(x_2) = 1.341\,641$
$x_3 = 0.6$	$y_3 = 1.483\,281$	$y(x_3) = 1.483\,240$
$x_4 = 0.8$	$y_4 = 1.612\,514$	$y(x_4) = 1.612\,452$
$x_5 = 1.0$	$y_5 = 1.732\,142$	$y(x_5) = 1.732\,051$

可见,对四阶 R－K 公式即使取步长 $h = 0.2$ 计算,其计算结果也比一阶和二阶方法精度好得多.

8.6 一阶常微分方程组初值问题的数值解法

8.6.1 一阶常微分方程组的初值问题

联立 n 个一阶常微分方程,给定初始条件即形成该方程组的初值问题

$$\begin{cases} y_i'(x) = f_i(x, y_1(x), \cdots, y_n(x)), \quad x \in [a,b]; \\ y_i(x_0) = y_{i0}, \quad i = 1,2,\cdots,n. \end{cases} \tag{8-27}$$

其中,$y_i(x)(i = 1,2,\cdots,n)$ 均为定义在区间 $[a,b]$ 上的一元实值函数;$f_i(x, y_1(x), y_2(x), \cdots, y_n(x))$ 是关于 $x, y_1, y_2, \cdots, y_n$ 的多元实值函数.

记 $$\boldsymbol{y} = (y_1, y_2, \cdots, y_n)^{\mathrm{T}}, \quad \boldsymbol{y}_0 = (y_{10}, y_{20}, \cdots, y_{n0})^{\mathrm{T}},$$
$$\boldsymbol{f} = (f_1, f_2, \cdots, f_n)^{\mathrm{T}},$$

则初值问题式(8 − 27) 可表示为向量形式

$$\begin{cases} \boldsymbol{y}' = \boldsymbol{f}(x, \boldsymbol{y}), \quad x \in [a,b], \\ \boldsymbol{y}(x_0) = \boldsymbol{y}_0. \end{cases}$$

例 8 下面给出一阶常微分方程的一些例子:

(1)
$$\begin{cases} y' = -3y + z + e^{2x}, \\ z' = -y - 5z + e^{x}, \\ y(x_0) = y_0, \quad z(x_0) = z_0. \end{cases}$$

其中,$y(x)$,$z(x)$ 定义在区间 $[a,b]$ 上,称为常系数线性非齐次方程组的初值问题.

(2)
$$\begin{cases} x'(t) = x + y, \\ y'(t) = x + y + t, \\ x(t_0) = x_0, \quad y(t_0) = y_0. \end{cases}$$

其中,$x(t)$,$y(t)$ 定义在区间 $[a,b]$ 上,称为常系数线性非齐次方程组的初值问题.

(3)
$$\begin{cases} x'(t) = 1 - \dfrac{1}{y}, \\ y'(t) = \dfrac{1}{x - t}. \end{cases}$$

此方程组为一阶非线性方程组.

(4) 洛特卡 − 沃尔泰拉(Lotka − Volterra) 方程

$$\begin{cases} \dfrac{\mathrm{d}y}{\mathrm{d}x} = y + \alpha yz, \\ \dfrac{\mathrm{d}z}{\mathrm{d}x} = -z + \beta yz. \end{cases}$$

其中,x 表示时间;y 表示被捕食者动物群体的数量(如野兔);z 表示捕食者动物群体的数量(如狐狸);$\alpha < 0, \beta > 0$ 为参数. Lotka − Volterra 方程亦为一阶非线性方程组.

8.6.2 一阶常微分方程组的数值解法

以含有两个函数的方程组为例. 设方程组

$$
\begin{cases}
y'(x) = f_1(x,y,z), \\
z'(x) = f_2(x,y,z), \\
y(x_0) = y_0, \\
z(x_0) = z_0, \quad x \in [a,b].
\end{cases}
$$

的数值解计算表格如表 8 － 10 所示.

表 8 － 10

x_i	x_0	$x_1 = x_0 + h$	…	$x_N = x_0 + Nh$
y_i	y_0	$y_1 \approx y(x_1)$	…	$y_N \approx y(x_N)$
z_i	z_0	$z_1 \approx z(x_1)$	…	$z_N \approx z(x_N)$

1) 用显式 Euler 公式求解

Euler 公式
$$
\begin{cases}
y_{n+1} = y_n + h f_1(x_n, y_n, z_n), \\
z_{n+1} = z_n + h f_2(x_n, y_n, z_n).
\end{cases}
$$

式中，$n = 0,1,\cdots,N-1$. 该式可改写成

$$
\begin{cases}
y_{n+1} = y_n + hK, \\
z_{n+1} = z_n + hL, \\
K = f_1(x_n, y_n, Z_n), \\
L = f_2(x_n, y_n, z_n).
\end{cases}
$$

用显式 Euler 公式求解的计算表格见表 8 － 11.

表 8 － 11

x_n	y_n	z_n	$K = f_1(x_n, y_n, z_n)$	$L = f_2(x_n, y_n, z_n)$
x_0	y_0	z_0	$K = f_1(x_0, y_0, z_0)$	$L = f_2(x_0, y_0, z_0)$
$x_1 = x_0 + h$	$y_1 = y_0 + hK$	$z_1 = z_0 + hL$	$K = f_1(x_1, y_1, z_1)$	$L = f_2(x_1, y_1, z_1)$
⋮	⋮	⋮	⋮	⋮
$x_N = x_{N-1} + h$	$y_N = y_{N-1} + hK$	$z_N = z_{N-1} + hL$		

2) 用改进的 Euler 公式求解

$$
\begin{cases}
y_{n+1} = y_n + \dfrac{h}{2}(K_1 + K_2), \\
z_{n+1} = z_n + \dfrac{h}{2}(L_1 + L_2), \quad n = 0,1,\cdots,N-1.
\end{cases}
$$

其中

$$
\begin{cases}
K_1 = f_1(x_n, y_n, z_n), \\
L_1 = f_2(x_n, y_n, z_n);
\end{cases}
$$

$$
\begin{cases}
K_2 = f_1(x_n + h, y_n + hK_1, z_n + hL_1), \\
L_2 = f_2(x_n + h, y_n + hK_1, z_n + hL_1).
\end{cases}
$$

改进的 Euler 公式求解的计算表格见表 8 － 12.

请读者就例 8 中的四个初值问题，分别用 Euler 方法及改进的 Euler 方法写出求解公式及计算表格.

在例 8 的第四个初值问题中，若给定 $\alpha = 0.01, \beta = 0.02$，取初值 $y(0) - 20, z(0) = 20$，$x \in [0,15]$，用改进的 Euler 公式求解，则有

$$\begin{cases} y_{n+1} = y_n + \dfrac{h}{2}(K_1 + K_2), \\ z_{n+1} = z_n + \dfrac{h}{2}(L_1 + L_2); \end{cases}$$

$$\begin{cases} K_1 = y_n - 0.01 y_n z_n, \\ L_1 = -z_n + 0.02 y_n z_n; \end{cases}$$

$$\begin{cases} K_2 = (y_n + hK_1) - 0.01(y_n + hK_1)(z_n + hL_1), \\ L_2 = -(z_n + hL_1) + 0.02(y_n + hK_1)(z_n + h_1). \end{cases}$$

取 $h = 0.1, n = 0,1,2,\cdots,14$，计算结果如图 8 − 4 所示.

表 8 − 12

x_n	y_n	z_n	$K_i(i=1,2)$	$L_i(i=1,2)$	K^*, L^*
x_0 $x_0 + h$	y_0 $y_0 + hK_1$	z_0 $z_0 + hL_1$	$K_1 = f_1(x_0, y_0, z_0)$ $K_2 = f_1(x_0 + h,$ $y_0 + hK_1, z_0 + hL_1)$	$L_1 = f_2(x_0, y_0, z_0)$ $L_2 = f_2(x_0 + h,$ $y_0 + hK_1, z_0 + hL_1)$	$K^* = \frac{1}{2}(K_1 + K_2)$ $L^* = \frac{1}{2}(L_1 + L_2)$
$x_1 = x_0 + h$ $x_1 + h$	$y_1 = y_0 + hK^*$ $y_1 + hK_1$	$z_1 = z_0 + hL^*$ $z_1 + hL_1$	$K_1 = f_1(x_1, y_1, z_1)$ $K_2 = f_1(x_1 + h,$ $y_1 + hK_1, z_1 + hL_1)$	$L_1 = f_2(x_1, y_1, z_1)$ $L_2 = f_2(x_1 + h,$ $y_1 + hK_1, z_1 + hL_1)$	$K^* = \frac{1}{2}(K_1 + K_2)$ $L^* = \frac{1}{2}(L_1 + L_2)$
$x_2 = x_1 + h$ ⋮ $x_N = x_{N-1} + h$	$y_2 = y_1 + hK^*$ ⋮ $y_N = y_{N-1} + hK^*$	$z_2 = z_1 + hL^*$ ⋮ $z_N = z_{N-1} + hL^*$	⋮	⋮	⋮

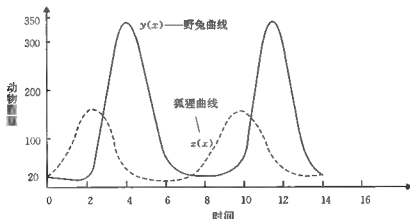


图8 4

图 8－4 为被捕食者(野兔)和捕食者(狐狸)两个种群的数量变化规律示意图. 初始时刻野兔和狐狸数量均为 20，当狐狸数量增加时，野兔数量开始减少；当狐狸数量达到峰值后便开始递减，此时野兔数量回升；野兔数量的峰值大约为 340，当达到峰值后其数量再次减少，而狐狸数量的峰值大约为 160. 两种动物的数量都呈现周期性变化，这就是捕食者与被捕食者系统的振荡现象.

8.7 高阶方程初值问题的数值解法

高阶方程的初值问题如下：

$$\begin{cases} y^{(n)}(x) = f(x, y, y', \cdots, y^{(n-1)}), \\ y(x_0) = y_0, y'(x_0) = y_0', \cdots, y^{(n-1)}(x_0) = y_0^{(n-1)}. \end{cases}$$

令

$$\begin{cases} y(x) = y_1(x), \\ y_1'(x) = y_2(x), & (y'), \\ y_2'(x) = y_3(x), & (y''), \\ \quad\vdots \\ y_{n-1}'(x) = y_n(x), & (y^{(n-1)}), \\ y_n'(x) = y^{n}(x) = f(x, y_1, \cdots, y_n). \end{cases}$$

由此形成 n 个一阶方程组成的一阶微分方程组的初值问题：

$$\begin{cases} y_1'(x) = y_2(x), & (f_1 = y_2), \\ y_2'(x) = y_3(x), & (f_2 = y_3), \\ \quad\vdots \\ y_{n-1}'(x) = y_n(x), & (f_{n-1} = y_n), \\ y_n'(x) = f(x, y_1, \cdots, y_n) = f_n. \end{cases}$$

$$y_i(x_0) = y_{i0}, \qquad i = 1, 2, \cdots, n.$$

以常用的二阶微分方程为例：

设
$$\begin{cases} y'' = f(x, y, y'), \\ y(x_0) = y_0, \quad y'(x_0) = y_0', \quad \in [a, b]. \end{cases}$$

令 $y(x) = y_1(x)$，形成

$$\begin{cases} y_1'(x) = y_2(x), \\ y_2'(x) = f(x, y_1, y_2), \end{cases}$$

$$y_1(x_0) = y_{10} = y_0, \quad y_2(x_0) = y_{20} = y_0',$$

或令 $y' = z$，形成

$$\begin{cases} y' = z, \\ z' = f(x, y, z), \end{cases}$$

$$y(x_0) = y_0, \quad z(x_0) = y'(x_0) = y_0'.$$

其数值解见表 8－13.

表 8－13

x_i	x_0	x_1	x_2	…	x_N
y_i	y_0	y_1	y_2	…	y_N
$z_i(y_i')$	z_0	z_1	z_2	…	z_n
	(y_0')	(y_1')	(y_2')	…	(y_n')

例 9 写出以下二阶微分方程的求解公式.

(1) $\begin{cases} y'' - 3y' + 2y = 0, \\ y(0) = 1, \quad y'(0) = 1. \end{cases}$

解 据 $y'' = -2y + 3y'$，令 $y' = z$，形成一阶微分方程组

$$\begin{cases} y' = z, \quad y(0) = 1, \\ z' = -2y + z, \quad z(0) = 1. \end{cases}$$

对上述一阶方程组的初值问题，分别应用 Euler 公式及改进的 Euler 公式.

①Euler 公式

$$\begin{cases} y_{n+1} = y_n + hK, \\ z_{n+1} = z_n + hL. \end{cases}$$

其中

$$\begin{cases} K = z_n, \\ L = -2y_n + 3z_n, \quad n = 0,1,\cdots,N. \end{cases}$$

② 改进的 Euler 公式

$$\begin{cases} y_{n+1} = y_n + \dfrac{h}{2}(K_1 + K_2), \\ z_{n+1} = z_n + \dfrac{h}{2}(L_1 + L_2). \end{cases}$$

其中

$$\begin{cases} K_1 = z_n, \\ L_1 = -2y_n + 3z_n; \end{cases}$$

$$\begin{cases} K_2 = z_n + hL_1, \\ L_2 = -2(y_n + hK_1) + 3(z_n + hL_1). \end{cases}$$

(2) $\begin{cases} y'' - 0.1(1 - y^2)y' + y = 0, \\ y(0) = 1, \quad y'(0) = 0. \end{cases}$

解 据 $y'' = 0.1(1 - y^2)y' - y$，令 $y' = z$，形成一阶微分方程组

$$\begin{cases} y' = z, \\ z' = 0.1(1 - y^2)z - y. \end{cases}$$

$$y(0) = 1, \quad z(0) = 0.$$

请读者对这一初值问题分别写出 Euler 公式及改进的 Euler 公式.

(3) $\begin{cases} x''(t) = -x/r^3, \\ y''(t) = -y/r^3. \end{cases}$

$x(0) = 0.4, \quad x'(0) = 0,$

$y(0) = 0, \quad y'(0) = 2.$

其中 $r = (x^2 + y^2)^{1/2}$.

解　令 $x' = z, y' = u$,形成一阶微分方程组

$$\begin{cases} x' = z, & x(0) = 0.4; \\ z' = -x/r^3, & z(0) = 0; \\ y' = u, & y(0) = 0; \\ u' = -y/r^3, & u(0) = 2. \end{cases}$$

①Euler 公式

$$\begin{cases} x_{n+1} = x_n + hK, \\ z_{n+1} = z_n + hL, \\ y_{n+1} = y_n + hM, \\ u_{n+1} = u_n + hN. \end{cases}$$

其中

$$\begin{cases} K = z_n, \\ L = \dfrac{-x_n}{(x_n^2 + y_n^2)^{3/2}}, \\ M = u_n, \\ N = \dfrac{-y_n}{(x_n^2 + y_n^2)^{3/2}}. \end{cases}$$

② 改进的 Euler 公式

$$\begin{cases} x_{n+1} = x_n + \dfrac{h}{2}(K_1 + K_2), \\ z_{n+1} = z_n + \dfrac{h}{2}(L_1 + L_2), \\ y_{n+1} = y_n + \dfrac{h}{2}(M_1 + M_2), \\ u_{n+1} = u_n + \dfrac{h}{2}(N_1 + N_2). \end{cases}$$

其中

$$\begin{cases} K_1 = z_n, \\ L_1 = -\dfrac{x_n}{(x_n^2 + y_n^2)^{3/2}}, \\ M_1 = u_n, \\ N_1 = -\dfrac{y_n}{(x_n^2 + y_n^2)^{3/2}}. \end{cases}$$

$$\begin{cases} K_2 = z_n + hL_1, \\ L_2 = -\dfrac{x_n + hK_1}{[(x_n + hK_1)^2 + (y_n + hM_1)^2]^{3/2}}, \\ M_2 = u_n + hN_1, \\ N_2 = \dfrac{-y_n + hM_1}{[(x_n + hK_1)^2 + (y_n + hM_1)^2]^{3/2}}. \end{cases}$$

例 10(火箭飞行模型)　设一火箭点火发射时,火箭的质量为 1 360.8 kg,其中燃料的质量为 1 088.6 kg,此燃料消耗速率为 18.14 kg/s,并产生 311 148 N 的推力.

(1) 根据 Newton 第二定律导出火箭飞行位移 $s(t)$ 的微分方程.

(2) 应用数值方法求解.

解　(1) 设推力 $F = 311\,148$ N，t 时刻火箭的质量 $W = 1\,360.8 - 18.14t$，重力加速度 $g = 9.807\text{m/s}^2$，大气阻力 $= kV^2$，其中阻力系数 $k = 0.011\,9\text{kg}\cdot\text{s}^2/\text{m}$.

据 Newton 第二定律

$$ma = F - mg - kV^2.$$

将 $m = W/g$ 代入上式得

$$a = \frac{F}{m} - g - \frac{k}{m}V^2 = \frac{g}{W}F - g - \frac{gk}{W}V^2,$$

即

$$\frac{\mathrm{d}S^2}{\mathrm{d}t^2} = \frac{g}{W}F - g - \frac{gk}{W}\left(\frac{\mathrm{d}S}{\mathrm{d}t}\right)^2.$$

令 $V = \frac{\mathrm{d}S}{\mathrm{d}t}$，将上式化为一阶微分方程组

$$\begin{cases} S' = V, \\ V' = \frac{g}{W}F - g - \frac{gk}{W}V^2. \end{cases}$$

(2) 设初值 $S(0) = S'(0) = 0$，$t \in [0,15]$ 形成初值问题

$$\begin{cases} S' = V, \\ V' = \frac{g}{W}F - g - \frac{gk}{W}V^2, \quad t \in [0,15]. \end{cases}$$

其中

$$S(0) = 0, \quad V(0) = 0.$$

将时间区间 $[0,15]$ 分成 N 等份，请读者应用改进的 Euler 方法计算位移 S，速度 $\frac{\mathrm{d}S}{\mathrm{d}t}$，加速度 $\frac{\mathrm{d}^2S}{\mathrm{d}t^2}$，并给出这三个变量对时间的关系曲线.

8.8　单步法的收敛性和稳定性

8.8.1　引言

设一阶常微分方程的初值问题

$$\begin{cases} y' = f(x,y), \quad x \in [a,b], \\ y(a) = \eta. \end{cases} \tag{8-28}$$

求解式(8－28)单步方法的一般形式为

$$\begin{cases} y_{n+1} = y_n + h\Phi(x_n, y_n, h), \\ y_0 = \eta. \end{cases} \tag{8-29}$$

其中，$\Phi(x_n, y_n, h)$ 称为增量函数. 若用单步方法(8－29)的解 $y_n(n = 0,1,\cdots,N)$ 近似代替式(8－28)的解 $y(x_n)(n = 0,1,\cdots,N)$，即

$$y(x_n) \approx y_n,$$

则必须考虑以下两个问题.

1) 差分方程(8－29)解的收敛性问题

式(8－29)解的收敛性问题是指在不考虑初始数据 $y_0=\eta$ 的误差及方法的舍入误差的情况下，研究任意固定点 $x_n=a+nh$，式(8－29)的理论解 y_n 与式(8－28)的精确解 $y(x_n)$ 的逼近程度(当 $h\to 0$ 或 $n\to\infty$ 时).

事实上，对于任意固定的点 $x_n=a+nh$，x_n 既依赖于 h，又依赖于 n. 由于 x_n 固定，当 $h\to 0$ 时，有 $n\to\infty$. 若要考虑当 $h\to 0(n\to\infty)$ 时，$y_n^{(h)}$ 是否趋于 $y(x_n)$，即是说，式(8－29)解的收敛问题归结为判定极限

$$\lim_{\substack{h\to 0\\(n\to\infty)\\x_n=a+nh}} y_n^{(h)}=y(x_n)$$

对于区间 $[a,b]$ 内的所有点 x_n 是否成立，若成立则称式(8－29)是收敛的.

直接用定义检验单步方法的收敛性很困难，下面我们进一步介绍判别单步方法收敛性的充分条件.

定理 8.1 设

(1) 初值问题(8－28)所对应的单步方法(8－29)具有 p 阶精度；

(2) 增量函数 $\Phi(x,y,h)$ 在区域 $x\in[a,b]$，$y\in(-\infty,+\infty)$，$h\in[0,h_0]$ 中连续，且对变量 y 满足 Lipschitz 条件，即

$$|\Phi(x,y,h)-\Phi(x,\bar{y},h)|\leqslant L|y-\bar{y}|$$

在上述区域内成立；

(3) 初值 y_0 是精确的，即 $y_0=y(x_0)$.

单步方法(8－29)在 $[x_0,x_n]$ 上的整体截断误差为

$$y(x_n)-y_n=O(h^p). \tag{8－30}$$

(该定理证明从略)

2) 差分方程(8－29)解的稳定性问题

在研究收敛性问题时，我们忽略了初始数据误差及计算中的舍入误差的影响. 事实上任何一种数值方法都不能回避这一问题.

当 h 固定时，我们考虑差分方程的近似解 $\bar{y}_n$ 与理论解 y_n 的逼近问题，即 $|y_n-\bar{y}_n|$ 的大小，这就是所谓方法的稳定性问题.

8.8.2 单步方法的收敛性

定义 8.3 若对于任意固定的点 $x_n=a+nh$，式(8－29)的理论解 y_n 与式(8－28)的精确解 $y(x_n)$ 之间有下式成立：

$$\lim_{\substack{h\to 0\\(n\to\infty)}} y_n^{(h)}=y(x_n),$$

则称单步方法(8－29)在 x_n 处收敛.

根据定理 8.1 的结论自然会导出在满足定理 8.1 的条件下单步方法(8－29)对任意固定点 $x_n=x_0+nh$ 是收敛的.

事实上，由于 $y(x_n)-y_n=O(h^p)$，故对 $x_n=x_0+nh$ 有下式成立：

$$\lim_{\substack{h\to 0\\(n\to\infty)}}[y(x_n)-y_n]=0,$$

即 $$\lim_{\substack{h\to 0\\(n\to\infty)}} y_n = y(x_n).$$

由于 x_n 的任意性，式(8-29)在$[a,b]$上收敛.

例 11　验证 Euler 公式及改进的 Euler 公式的收敛性.

解　(1)Euler 公式

$$y_{n+1} = y_n + hf(x_n, y_n).$$

这里 $\Phi(x_n, y_n, h) = f(x_n, y_n)$. 据定理 8.1，当 $f(x,y)$ 满足关于 y 的 Lipschitz 条件时，Euler 公式在任意固定点 $x_n = x_0 + nh$ 处收敛.

(2) 改进的 Euler 公式

$$y_{n+1} = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_n + h, y_n + hf(x_n, y_n))].$$

这里 $$\Phi(x, y, h) = \frac{1}{2}[f(x,y) + f(x+h, y+hf(x,y))].$$

若 $$|f(x,y) - f(x,\bar{y})| \leqslant L|y-\bar{y}|,$$

则
$$\begin{aligned}
&|f(x+h, y+hf(x,y)) - f(x+h, \bar{y}+hf(x,\bar{y}))| \\
&\leqslant L|y+hf(x,y) - \bar{y} - hf(x,\bar{y})| \\
&= L|(y-\bar{y}) + h[f(x,y) - f(x,\bar{y})]| \\
&\leqslant L|y-\bar{y}| + Lh|f(x,y) - f(x,\bar{y})| \\
&\leqslant |y-\bar{y}| + L^2h|y-\bar{y}|.
\end{aligned}$$

所以
$$\begin{aligned}
&|\Phi(x,y,h) - \Phi(x,\bar{y},h)| \\
&\leqslant \frac{1}{2}[L|y-\bar{y}| + L|y-\bar{y}| + L^2h|y-\bar{y}|] \\
&= \frac{1}{2}L|y-\bar{y}|(2+Lh) \\
&= L(1+\frac{h}{2})(y-\bar{y}).
\end{aligned} \tag{8-31}$$

式(8-31)表明增量函数 $\Phi(x,y,h)$ 关于 y 的 Lipschitz 常数为

$$L_\Phi = L_f(1+\frac{h}{2}).$$

综上所述，只要 $f(x,y)$ 满足 Lipschitz 条件，则 $\Phi(x,y,h)$ 仍满足 Lipschitz 条件(只是 Lipschitz 常数不同罢了). 因此改进的 Euler 公式对任意固定点 $x_n = x_0 + nh$ 是收敛的.

对于其他的单步公式可以类似地验证其收敛性. 由于最后只涉及到对 $f(x,y)$ 要求满足 y 的 Lipschitz 条件，可见所有的单步方法其收敛性的条件是容易满足的.

8.8.3　单步方法的稳定性

在许多实用的数值方法中，计算公式都是递推的，通常总是取有限的固定步长进行计算，初值 y_0 就可能含有误差. 而在计算过程中又会发生舍入误差，某一步的误差传递能否受到控制是与步长的大小有关的. 如果在计算过程中这些误差不是逐步扩大，则称此方法具有绝对稳定性，否则称此方法是不稳定的.

收敛性和稳定性是两个不同的概念. 收敛性是反映数值方法本身的截断误差对计算

结果的影响，而稳定性是反映计算某一步中出现的误差(例如舍入误差)对计算结果的影响. 绝对稳定性是和步长密切相关的，对某一步长是稳定的数值公式，对另一步长可能就不稳定. 显然，只有既收敛而又稳定的数值公式才能在实际计算中得到应用.

一种数值方法的绝对稳定性与具体的方程有关，即它强烈依赖于微分方程的右端函数 $f(x,y)$. 为了讨论方便，常选用试验方程

$$\begin{cases} y' = \lambda y, \\ y(x_0) = y_0 \end{cases} \tag{8-32}$$

来讨论绝对稳定性问题. 为了保证数值方法绝对稳定，λ 取为负实数，同时对参数 λ 和步长 h 都要加以一定的限制. 我们称 λ 和 h 的允许范围 $(\lambda_0, 0)$，即当 $\lambda, h \in (\lambda_0, 0)$ 误差传递不扩大时为相应方法的绝对稳定区. 显然 λ_0 越小，方法越稳定，h 的选择余地也就越大.

下面以四阶 R－K 公式为例来讨论绝对稳定区. 对于方程(8－32)，由四阶 R－K 公式的古典形式可以推出

$$y_{i+1} = (1 + \lambda h + \frac{1}{2}\lambda^2 h^2 + \frac{1}{6}\lambda^3 h^3 + \frac{1}{24}\lambda^4 h^4) y_i.$$

如果在 y_0 上有一个扰动 δ_0，且假设其后的全部计算都是精确的，则在 y_{i+1} 处引起的误差为

$$\begin{aligned} \delta_{i+1} &= (1 + \lambda h + \frac{1}{2}\lambda^2 h^2 + \frac{1}{6}\lambda^3 h^3 + \frac{1}{24}\lambda^4 h^4)\delta_i \\ &= (1 + \lambda h + \frac{1}{2}\lambda^2 h^2 + \frac{1}{6}\lambda^3 h^3 + \frac{1}{24}\lambda^4 h^4)^{i+1}\delta_0. \end{aligned}$$

因此，为了使 y_{i+1} 处的扰动 δ_{i+1} 不超过 δ_0，应满足

$$y_{i+1} = \left| 1 + \lambda h + \frac{1}{2}\lambda^2 h^2 + \frac{1}{6}\lambda^3 h^3 + \frac{1}{24}\lambda^4 h^4 \right| \leqslant 1. \tag{8-33}$$

在 λh 平面上，由不等式(8－33)所确定的区域就是四阶 R－K 的绝对稳定区域. 其中

$$\left| 1 + \lambda h + \frac{1}{2}\lambda^2 h^2 + \frac{1}{6}\lambda^3 h^3 + \frac{1}{24}\lambda^4 h^4 \right| = 1$$

为绝对稳定区的边界. 如果绝对稳定区的边界为无限，则称相应的数值方法是无条件稳定的.

表 8－14 列出了一些方法的绝对稳定区间，"—"表示无绝对稳定区间.

表 8－14

方法	方法的阶	稳定区间
Euler 法	1	(－2,0)
隐式 Euler 法	1	(－∞,0)
Euler 中点法	2	—
梯形法	2	(－∞,0)
梯形公式预报－校正法	2	(－2,0)
二阶 R－K 法	2	(－2,0)
三阶 R－K 法	3	(－2.51,0)
四阶 R－K 法	4	(－2.78,0)

习题 8

*1. 已知初值问题

$$\begin{cases} y' = \dfrac{1}{1+x^2} - 2y^2, & 0 \leqslant x \leqslant 1, \\ y(0) = 0. \end{cases}$$

试用 Euler 法，取不同步长 0.1 和 0.2，分别计算到 $y(1)$，并与准确解 $y(x) = \dfrac{x}{1+x^2}$ 相比较. 近似值和准确值都取四位有效数字.

*2. 用改进的 Euler 公式和四阶经典 R－K 法，取步长 $h = 0.1$，计算下列初值问题的近似解，并与精确值比较，且二者均保留四位小数.

$$\begin{cases} y' = x + y, & 0 \leqslant x \leqslant 1, \\ y(0) = 1. \end{cases}$$

此问题的精确解为 $y(x) = -x - 1 + 2e^x$.

3. 用梯形法解初值问题

$$\begin{cases} y' + y = 0, \\ y(0) = 1. \end{cases}$$

并证明其近似解为

$$y_n = \left(\frac{2-h}{2+h}\right)^n,$$

且当 $h \to 0$ 时，它收敛于原初值问题的准确解 $y = e^x$.

4. 利用 Euler 方法计算积分

$$\int_0^x e^{t^2} dt$$

在点 $x = 0.5, 1, 1.5, 2$ 处的近似值.

5. 证明对任意参数 t，下列 R－K 公式是二阶的：

$$\begin{cases} y_{n+1} = y_n + \dfrac{h}{2}(k_2 + k_3), \\ k_1 = f(x_n, y_n), \\ k_2 = f(x_n + th, y_n + thk_1), \\ k_3 = f(x_n + (1-t)h, y_n + (1-t)hk_1). \end{cases}$$

6. 确定下列差分公式的阶及截断误差的首项：

$$y_{n+1} = 4y_n - 3y_{n-1} - 2hf(x_{n-1}, y_{n-1}).$$

注 打 * 号的题目在本书附录 Ⅱ 作了详解.

*第 9 章　多元回归分析与趋势面分析简介

9.1　多元线性回归分析

9.1.1　引言

客观世界中普遍存在着变量之间的关系. 当其变量之间存在确定性关系，即数值对应关系(或称函数关系)时，可用第 6 章中讨论的插值方法和曲线拟合的方法处理实测数据. 但是在科学或工程实践中，变量之间的关系大多是属于非确定性的. 例如，人的身高和体重之间的关系，人的血压和年龄之间的关系，禾苗的生长量与施肥量的关系，等等. 这些关系不能用传统的函数关系来表示，只能研究它们"统计意义"上的关系，即研究随机现象中数据之间的内在规律性. 常用的统计方法有多元回归分析、趋势面分析、判别分析、聚类分析和降维技术等. 本节将简要地介绍多元线性回归分析和趋势面分析，所涉及到的一些统计量的计算公式，只介绍它们的作用，而这些公式的概率统计原理，请读者参阅参考书目[6]，[7].

9.1.2　多元线性回归的数学模型

回归分析方法是在大量试验观测数据的基础上，找出变量之间的内部规律性，从而定量地建立一个随机变量和另外多个随机变量之间的内部关系(规律性)，即是建立它们之间统计关系的数学表达式. 简单地说就是"研究一个变量和其他变量之间的统计关系"，这种统计关系常用回归方程表示.

回归分析中被回归的变量 y 称为因变量，影响 y 变化的其他变量 $x_1, x_2, \cdots, x_m$ 称为自变量. 如果自变量只有一个，称为一元回归；如果自变量有两个及以上，称为多元回归. 如果 y 与 $x_1, x_2, \cdots, x_m$ 之间的关系是线性关系，则称为线性回归；否则称为非线性回归.

设随机变量 y 与 m 个自变量存在线性关系

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m + \varepsilon. \tag{9-1}$$

式(9－1)称为回归方程，其中 $\beta_0, \beta_1, \cdots, \beta_m$ 称为回归系数；ε 为随机变量，亦称为随机误差，它可理解为 y 无法用 $x_1, x_2, \cdots, x_m$ 表示的其他各种随机因素所造成的误差. 据概率统计原理，随机变量 ε, y 均服从正态分布，即 $\varepsilon \sim N(0, \sigma^2)$，$y \sim N(E(y), \sigma^2)$.

下面对变量 $x_1, x_2, \cdots, x_m, y$ 测取 n 组样本数据，如表 9－1 所示.

表 9－1

x_1	x_2	$\cdots$	x_m	y
x_{11}	x_{12}	$\cdots$	x_{1m}	y_1
x_{21}	x_{22}	$\cdots$	x_{2m}	y_2
$\vdots$	$\vdots$		$\vdots$	$\vdots$
x_{n1}	x_{n2}	$\cdots$	x_{nm}	y_n

将表 9－1 数据代入式(9－1)，于是有

$$\begin{cases} y_1 = \beta_0 + \beta_1 x_{11} + \beta_2 x_{12} + \cdots + \beta_m x_{1m} + \varepsilon_1, \\ y_2 = \beta_0 + \beta_1 x_{21} + \beta_2 x_{22} + \cdots + \beta_m x_{2m} + \varepsilon_2, \\ \quad \vdots \\ y_n = \beta_0 + \beta_1 x_{n1} + \beta_2 x_{n2} + \cdots + \beta_m x_{nm} + \varepsilon_n. \end{cases} \tag{9-2}$$

其中，$\beta_0, \beta_1, \cdots, \beta_m$ 为 $m+1$ 个待定参数；$\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_n$ 为 n 个相互独立且服从同一正态分布 $N(0, \sigma^2)$ 的随机变量. 式(9－2) 称为 m 元线性回归数学模型，其亦可写成矩阵形式. 设

$$\boldsymbol{X} = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1m} \\ 1 & x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nm} \end{pmatrix},$$

$$\boldsymbol{\beta} = (\beta_0, \beta_1, \cdots, \beta_m)^{\mathrm{T}}, \quad \boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_n)^{\mathrm{T}}, \quad \boldsymbol{y} = (y_1, y_2, \cdots, y_n)^{\mathrm{T}},$$

则式(9－2) 可表示为

$$\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\varepsilon}. \tag{9-3}$$

式(9－3) 称为多元线性回归模型的矩阵形式.

9.1.3 回归模型中的参数估计

设 $b_0, b_1, \cdots, b_m$ 分别是式(9－2) 中待定参数 $\beta_0, \beta_1, \cdots, \beta_m$ 的最小二乘估计值，于是有

$$\hat{y} = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_m x_m. \tag{9-4}$$

对于每一个样本数据 $(x_{i1}, x_{i2}, \cdots, x_{im})$，$i = 1, 2, \cdots, n$ 对应一个回归值 $\hat{y}_i$，即是

$$\hat{y}_i = b_0 + b_1 x_{i1} + b_2 x_{i2} + \cdots + b_m x_{im}, \quad i = 1, 2, \cdots, n.$$

为了估计 $b_i (i = 0, 1, \cdots, m)$，现令

$$b_0 + b_1 x_{i1} + b_2 x_{i2} + \cdots + b_m x_{im} = y_i, \quad i = 1, 2, \cdots, n. \tag{9-5}$$

式(9－5) 即是超定方程组($n \gg m$)，它的矩阵形式为

$$\boldsymbol{Xb} = \boldsymbol{y}. \tag{9-6}$$

其中

$$\boldsymbol{b} = (b_0, b_1, \cdots, b_m)^{\mathrm{T}}.$$

而超定方程组(9－5) 或(9－6) 的最小二乘解可通过求解对应的法方程组

$$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Xb} = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} \tag{9-7}$$

得到. 式(9－7) 中

$$X^TX = \begin{pmatrix} n & \sum_{i=1}^{n} x_{i1} & \sum_{i=1}^{n} x_{i2} & \cdots & \sum_{i=1}^{n} x_{im} \\ \sum_{i=1}^{n} x_{i1} & \sum_{i=1}^{n} x_{i1}^2 & \sum_{i=1}^{n} x_{i1}x_{i2} & \cdots & \sum_{i=1}^{n} x_{i1}x_{im} \\ \sum_{i=1}^{n} x_{i2} & \sum_{i=1}^{n} x_{i2}x_{i1} & \sum_{i=1}^{n} x_{i2}^2 & \cdots & \sum_{i=1}^{n} x_{i2}x_{im} \\ \vdots & \vdots & \vdots & & \vdots \\ \sum_{i=1}^{n} x_{im} & \sum_{i=1}^{n} x_{im}x_{i1} & \sum_{i=1}^{n} x_{im}x_{i2} & \cdots & \sum_{i=1}^{n} x_{im}^2 \end{pmatrix},$$

$$X^Ty = \left(\sum_{i=1}^{n} y_i, \sum_{i=1}^{n} x_{i1}y_i, \sum_{i=1}^{n} x_{i2}y_i, \cdots, \sum_{i=1}^{n} x_{im}y_i\right)^T.$$

例如，当 $m = 2$ 时，则

$$\hat{y} = b_0 + b_1x_1 + b_2x_2,$$

$$X^TX = \begin{pmatrix} n & \sum_{i=1}^{n} x_{i1} & \sum_{i=1}^{n} x_{i2} \\ \sum_{i=1}^{n} x_{i1} & \sum_{i=1}^{n} x_{i1}^2 & \sum_{i=1}^{n} x_{i1}x_{i2} \\ \sum_{i=1}^{n} x_{i2} & \sum_{i=1}^{n} x_{i2}x_{i1} & \sum_{i=1}^{n} x_{i2}^2 \end{pmatrix},$$

$$X^Ty = \left(\sum_{i=1}^{n} y_i, \sum_{i=1}^{n} x_{i1}y_i, \sum_{i=1}^{n} x_{i2}y_i\right)^T.$$

矩阵 X^TX 及右端向量 X^Ty 的元素可列表进行计算(见表 9 − 2). 解三元一次方程 $X^TXb = X^Ty$ 可得回归参数的估计值 b_0, b_1, b_2.

表 9 − 2

x_{i1}	x_{i2}	$x_{i1}x_{i2}$	x_{i1}^2	x_{i2}^2	y_i	$x_{i1}y_i$	$x_{i2}y_i$
x_{11}	x_{12}	$x_{11}x_{12}$	x_{11}^2	x_{12}^2	y_1	$x_{11}y_1$	$x_{12}y_1$
x_{21}	x_{22}	$x_{21}x_{22}$	x_{21}^2	x_{22}^2	y_2	$x_{21}y_2$	$x_{22}y_2$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
x_{n1}	x_{n2}	$x_{n1}x_{n2}$	x_{n1}^2	x_{n2}^2	y_n	$x_{n1}y_n$	$x_{n1}y_n$
$\sum_{i=1}^{n} x_{i1}$	$\sum_{i=1}^{n} x_{i2}$	$\sum_{i=1}^{n} x_{i1}x_{i2}$	$\sum_{i=1}^{n} x_{i1}^2$	$\sum_{i=1}^{n} x_{i2}^2$	$\sum_{i=1}^{n} y_i$	$\sum_{i=1}^{n} x_{i1}y_i$	$\sum_{i=1}^{n} x_{i2}y_i$

当 $m = 3$ 时，则

$$\hat{y} = b_0 + b_1x_1 + b_2x_2 + b_3x_3,$$

$$
X^{T}X=\begin{pmatrix}
n & \sum_{i=1}^{n}x_{i1} & \sum_{i=1}^{n}x_{i2} & \sum_{i=1}^{n}x_{i3}\\
\sum_{i=1}^{n}x_{i1} & \sum_{i=1}^{n}x_{i1}^{2} & \sum_{i=1}^{n}x_{i1}x_{i2} & \sum_{i=1}^{n}x_{i1}x_{i3}\\
\sum_{i=1}^{n}x_{i2} & \sum_{i=1}^{n}x_{i2}x_{i1} & \sum_{i=1}^{n}x_{i2}^{2} & \sum_{i=1}^{n}x_{i2}x_{i3}\\
\sum_{i=1}^{n}x_{i3} & \sum_{i=1}^{n}x_{i3}x_{i1} & \sum_{i=1}^{n}x_{i3}x_{i2} & \sum_{i=1}^{n}x_{i3}^{2}
\end{pmatrix},
$$

$$
X^{T}y=\left(\sum_{i=1}^{n}y_i,\sum_{i=1}^{n}x_{i1}y_i,\sum_{i=1}^{n}x_{i2}y_i,\sum_{i=1}^{n}x_{i3}y_i\right)^{T}.
$$

$X^{T}X$ 及 $X^{T}y$ 中各元素的计算见表 9－3. 解四元一次方程组 $X^{T}Xb=X^{T}y$ 可得回归参数的估计值 b_0,b_1,b_2,b_3.

表 9－3

x_{i1}	x_{11}	…	x_{n1}	$\sum_{i=1}^{n}x_{i1}$	y_i	y_1	…	$x_{n1}y_1$	$\sum_{i=1}^{n}y_i$
x_{i2}	x_{12}	…	x_{n2}	$\sum_{i=1}^{n}x_{i2}$	$x_{i1}y_i$	$x_{11}y_1$	…	y_n	$\sum_{i=1}^{n}x_{i1}y_i$
x_{i3}	x_{13}	…	x_{n3}	$\sum_{i=1}^{n}x_{i3}$	$x_{i2}y_i$	$x_{12}y_2$	…	$x_{n2}y_2$	$\sum_{i=1}^{n}x_{i2}y_i$
$x_{i1}x_{i2}$	$x_{11}x_{12}$	…	$x_{n1}x_{n2}$	$\sum_{i=1}^{n}x_{i1}x_{i2}$	$x_{i3}y_i$	$x_{13}y_1$	…	$x_{n3}y_3$	$\sum_{i=1}^{n}x_{i3}y_i$
$x_{i1}x_{i3}$	$x_{11}x_{13}$	…	$x_{n1}x_{n3}$	$\sum_{i=1}^{n}x_{i1}x_{i3}$					
$x_{i2}x_{i3}$	$x_{21}x_{23}$	…	$x_{n2}x_{n3}$	$\sum_{i=1}^{n}x_{i2}x_{i3}$					
x_{i1}^{2}	x_{11}^{2}	…	x_{n1}^{2}	$\sum_{i=1}^{n}x_{i1}^{2}$					
x_{i2}^{2}	x_{12}^{2}	…	x_{n2}^{2}	$\sum_{i=1}^{n}x_{i2}^{2}$					
x_{i3}^{2}	x_{13}^{2}	…	x_{n3}^{2}	$\sum_{i=1}^{n}x_{i3}^{2}$					

下面介绍估计参数 $b_0,b_1,\cdots,b_m$ 的另一种计算方案.

令
$$
\tilde{X}=\begin{pmatrix}
\tilde{x}_{11} & \tilde{x}_{12} & \cdots & \tilde{x}_{1m}\\
\tilde{x}_{21} & \tilde{x}_{22} & \cdots & \tilde{x}_{2m}\\
\vdots & \vdots & & \vdots\\
\tilde{x}_{n1} & \tilde{x}_{n2} & \cdots & \tilde{x}_{nm}
\end{pmatrix},
$$

$$
\tilde{y}=(\tilde{y}_1,\tilde{y}_2,\cdots,\tilde{y}_n)^{T}.
$$

其中 $\tilde{x}_{ij}=x_{ij}-\bar{x}_j(i=1,2,\cdots,n;j=1,2,\cdots,m)$,

$\tilde{y}_i=y_i-\bar{y}(i=1,2,\cdots,n)$,

$\bar{x}_j=\frac{1}{n}\sum_{k=1}^{n}x_{kj},\bar{y}=\frac{1}{n}\sum_{i=1}^{n}y_i$.

于是可以构建相应的超定方程组

$$\widetilde{X}b = y. \tag{9-8}$$

其中 $b = (b_1, b_2, \cdots, b_m)^T$.

与(9－8)对应的法方程组为

$$\widetilde{X}^T\widetilde{X}b = \widetilde{X}^T y. \tag{9-9}$$

记 $C = \widetilde{X}^T\widetilde{X}, d = \widetilde{X}^T y$，则有 $Cb = d$，其中系数矩阵 C 及右端向量 d 的元素计算公式如下：

$$C_{ij} = \sum_{k=1}^{n}(x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j), \quad i, j = 1, 2, \cdots, m;$$

$$d_j = \sum_{k=1}^{n}(x_{kj} - \bar{x}_j)(y_k - \bar{y}), \quad j = 1, 2, \cdots, m.$$

由法方程组(9－9)解得

$$b = (b_1, b_2, \cdots, b_m)^T.$$

又由回归方程(9－4)可推得

$$\bar{y} = b_0 + b_1\bar{x}_1 + b_2\bar{x}_2 + \cdots + b_m\bar{x}_m,$$

从而解出 b_0，且

$$b_0 = \bar{y} - (b_1\bar{x}_1 + b_2\bar{x}_2 + \cdots + b_m\bar{x}_m).$$

于是得回归方程参数的全部估计值

$$b_0, b_1, b_2, \cdots, b_m.$$

显然，法方程组(9－9)比法方程组(9－7)降了一阶，这将对以后在检验回归变量的显著性以及估计回归值的预测区间时，计算 C 的逆阵起到很大的作用.

例 1　某养猪场估算猪的毛重，测得 14 头猪的体长 x_1 和胸围 x_2 与体重 y 的数据如表 9－4 所示. 求：

(1) 建立 y 与 x_1、x_2 的线性回归方程；

(2) 现有某头猪体长为 95cm，胸围为 80cm，试测其体重(kg).

表 9－4

序号	体长 x_1(cm)	胸围 x_2(cm)	体重 y(kg)
1	41	49	28
2	45	58	39
3	51	62	41
4	52	71	44
5	59	62	43
6	62	74	50
7	69	71	51
8	72	74	57
9	78	79	63
10	80	84	66
11	90	85	70
12	92	94	76
13	98	91	80
14	103	95	84

解 设随机变量 y 与 x_1、x_2 存在线性关系，即

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon,$$

则参数估计式为 $$\hat{y} = b_0 + b_1 x_1 + b_2 x_2.$$

将表 9－4 中 14 组样本数据代人上式，得参数估计值满足的超定方程组

$$y_i = b_0 + b_1 x_{i1} + b_2 x_{i2}, \quad i = 1,2,\cdots,14,$$

其矩阵形式为

$$Xb = y.$$

其中 $$X = \begin{pmatrix} 1 & x_{11} & x_{12} \\ 1 & x_{21} & x_{22} \\ \vdots & \vdots & \vdots \\ 1 & x_{14,1} & x_{14,2} \end{pmatrix} = \begin{pmatrix} 1 & 41 & 49 \\ 1 & 45 & 58 \\ \vdots & \vdots & \vdots \\ 1 & 103 & 95 \end{pmatrix},$$

$$b = (b_1, b_2, \cdots, b_m)^{\mathrm{T}}, \quad y = (28, 39, \cdots, 84)^{\mathrm{T}}.$$

求超定方程组 $Xb = y$ 的最小二乘解，对应的法方程组为

$$X^{\mathrm{T}}Xb = X^{\mathrm{T}}y.$$

据表 9－2，列表 9－5 计算法方程组的系数矩阵 $X^{\mathrm{T}}X$ 及右端向量 $X^{\mathrm{T}}y$ 的元素.

表 9－5

i	x_{i1}	x_{i2}	$x_{i1}x_{i2}$	x_{i1}^2	x_{i2}^2	y_i	$x_{i1}y_i$	$x_{i2}y_i$
1	41	49	41×49	41^2	49^2	28	41×28	49×28
2	45	58	45×58	45^2	58^2	39	45×39	58×39
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
14	103	95	103×95	103^2	95^2	84	103×84	95×84
$\sum$	992	1182	77829	75542	81151	792	60520	62380

由此可得 $$X^{\mathrm{T}}X = \begin{pmatrix} 14 & 992 & 1182 \\ 992 & 75542 & 77829 \\ 1182 & 77829 & 81151 \end{pmatrix},$$

$$X^{\mathrm{T}}y = (792, 60520, 62380)^{\mathrm{T}}.$$

解法方程组 $X^{\mathrm{T}}Xb = X^{\mathrm{T}}y$ 得参数向量的估计值

$$b = (b_1, b_2, \cdots, b_m)^{\mathrm{T}} = (-16.01, 0.522, 0.475)^{\mathrm{T}},$$

于是所求回归方程为

$$\hat{y} = -16.01 + 0.522x_1 + 0.475x_2. \qquad (9-10)$$

回归值 $\hat{y}_i$ 与实测值 y_i 以及回归值 $\hat{y}_i$ 与均值 $\bar{y}$ 的比较如表 9－6 所示.

表9 6

y_i	$\hat{y}_i$	$y_i-\hat{y}_i$	$\hat{y}_i-\bar{y}$	y_i	$\hat{y}_i$	$y_i-\hat{y}_i$	$\hat{y}_i-\bar{y}$
28	28.6660	−0.6660	−27.9054	57	56.7230	0.2770	0.1516
39	35.0290	3.9710	−21.5424	63	62.2300	0.7700	5.6585
41	40.0610	0.9390	−16.5104	66	65.6490	0.3510	9.0776
44	44.8580	−0.8580	−11.7134	70	71.3440	−1.3440	14.7726
43	44.2370	−1.2370	−12.3344	76	76.6630	−0.6630	20.0916
50	51.5030	−1.5030	−5.0684	80	78.3700	1.6300	21.7986
51	53.7320	−2.7320	−2.8394	84	82.8800	1.1200	26.3086

回归平面与实测点效果比较见图 9 − 1.

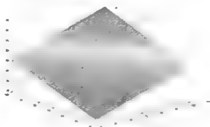


图 9 − 1

预测:现有某头猪体长为95cm,胸围为80cm,即 $x_1=95, x_2=80$,据式(9 − 10)计算得 $\hat{y}=71.58$(kg).

本例是一个二元线性回归问题,一元线性回归(一个自变量)及多元线性回归(自变量 ≥2) 可用同样的方法求得. 在讨论回归分析时,我们只是论及其方法,回归效果的分析留待后面作进一步讨论.

例 2　据经验,某反应器的产量 y 与原料中两种成分的含量 x_1、x_2 有线性关系. 现使用两种不同的催化剂 D_1、D_2,各做 6 次试验,取得表 9 − 7 所列数据. 试建立 y 与 x_1、x_2 的线性回归方程.

表 9 − 7

试验号	D_1			D_2		
	x_1	x_2	y	x_1	x_2	y
1	130	200	7.5	136	215	6.2
2	136	220	4.2	137	250	7.0

续表 9 7

试验号	D_1			D_2		
	x_1	x_2	y	x_1	x_2	y
3	140	215	1.5	136	180	5.5
4	138	265	3.7	138	240	5.6
5	134	235	5.3	139	220	4.6
6	142	260	1.2	141	260	3.9
均值	$\bar{x}_1=136.7$	$\bar{x}_2=232.5$	$\bar{y}=3.9$	$\bar{x}_1=137.8$	$\bar{x}_2=227.5$	$\bar{y}=5.5$

解 本题采用另一种方案估计回归方程中的参数.

(1) 就试验 D_1 的样本数据形成

$$\widetilde{X}=\begin{pmatrix}-6.7 & -32.5\\ -0.7 & -12.5\\ 3.3 & -17.5\\ 1.3 & 32.5\\ -2.7 & 2.5\\ 5.3 & 27.5\end{pmatrix},\quad \boldsymbol{y}=(3.6,0.3,-2.4,-0.2,1.4,-2.7)^{\mathrm{T}}.$$

设估计参数向量为

$$\boldsymbol{b}=(b_0,b_1,b_2)^{\mathrm{T}},$$

并令

$$\tilde{\boldsymbol{b}}=(b_1,b_2)^{\mathrm{T}}.$$

建立超定方程组

$$\widetilde{X}\tilde{\boldsymbol{b}}=\boldsymbol{y},$$

则对应的法方程组为

$$\widetilde{X}^{\mathrm{T}}\widetilde{X}\tilde{\boldsymbol{b}}=\widetilde{X}^{\mathrm{T}}\boldsymbol{y},$$

其中

$$\widetilde{X}^{\mathrm{T}}\widetilde{X}=\begin{pmatrix}93.34 & 252.5\\ 252.5 & 3337.5\end{pmatrix},\quad \widetilde{X}^{\mathrm{T}}\boldsymbol{y}=(-50.6,-15.6)^{\mathrm{T}},$$

其解为

$$\tilde{\boldsymbol{b}}=(-0.6655,0.0456)^{\mathrm{T}}.$$

再由

$$\bar{y}=b_0+b_1\bar{x}_1+b_2\bar{x}_2,$$

即

$$3.9=b_0-0.6655\times136.7+0.0456\times232.5,$$

得 $b_0=84.27$.

y 关于 x_1、x_2 的线性回归方程为

$$\hat{y}=84.27-0.6655x_1+0.0456x_2. \tag{9-11}$$

(2) 就试验 D_2 的样本数据形成

$$\widetilde{X}=\begin{pmatrix}-1.8 & -12.5\\ -0.8 & 22.5\\ -1.8 & -47.5\\ 0.2 & 12.5\\ 1.2 & -7.5\\ 3.2 & 32.5\end{pmatrix},\quad \boldsymbol{y}=(0.7,1.5,0.0,0.1,-0.9,-1.6)^{\mathrm{T}}.$$

超定方程组 $\widetilde{X}\tilde{b} = y$ 对应的法方程组为

$$\widetilde{X}^T\widetilde{X}\tilde{b} = \widetilde{X}^T y,$$

其中

$$\widetilde{X}^T\widetilde{X} = \begin{pmatrix} 18.84 & 187.5 \\ 187.5 & 4187.5 \end{pmatrix}, \quad \widetilde{X}^T y = (-8.64, -19)^T,$$

其解为 $b_1 = -0.4158, \quad b_2 = -0.0043.$

由 $\bar{y} = b_0 + b_1\bar{x}_1 + b_2\bar{x}_2$ 得

$$5.5 = b_0 - 0.4158 \times 137.8 - 0.0043 \times 227.5,$$

从而得 $b_0 = 63.78$.

y 关于 x_1、x_2 的线性回归方程为

$$\hat{y} = 63.78 - 0.4158x_1 - 0.0043x_2 \tag{9-12}$$

9.1.4 线性回归的效果检验

下面我们首先引入检验线性回归效果的几个统计量.

9.1.4.1 *总离差平方和 $S_总$*

$$S_总 = \sum_{i=1}^{n}(y_i - \bar{y})^2,$$

其中，$\bar{y} = \frac{1}{n}\sum_{i=1}^{n}\hat{y}_i$ 为回归变量$\hat{y}_i$ 的实测值 $y_i(i = 1,2,\cdots,n)$ 的均值. 下证：

$$S_总 = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2,$$

其中 $\hat{y}_i(i = 1,2,\cdots,n)$ 为回归变量$\hat{y}_i$ 的计算值.

事实上，

$$\begin{aligned} S_总 &= \sum_{i=1}^{n}[(y_i - \hat{y}_i) + (\hat{y}_i - \bar{y})]^2 \\ &= \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + 2\sum_{i=1}^{n}(y_i - \hat{y}_i)(\hat{y}_i - \bar{y}) + \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2. \end{aligned}$$

可以证明 $\sum_{i=1}^{n}(y_i - \hat{y}_i)(\hat{y}_i - \bar{y}) = 0$(证明从略)，故有

$$S_总 = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2.$$

9.1.4.2 *剩余平方和与回归平方和*

在总离差平方和 $S_总$ 中，称第一项平方和 $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ 为剩余平方和，记为 $S_剩$；称第二项平方和 $\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2$ 为回归平方和，记为 $S_回$，于是

$$S_总 = S_剩 + S_回. \tag{9-13}$$

$S_剩$ 是由试验误差以及其他因素引起的，它的大小反映试验误差及其他因素对试验结果的影响程度，其自由度为 $n - m - 1$；$S_回$ 反映了自变量 $x_1, x_2, \cdots, x_m$ 的变化所引起的对

y 的波动，其自由度为 m.

由式(9－13)看出，当 $S_{总}$ 确定时，$S_{剩}$ 越小，$S_{回}$ 就越大，即 $S_{回}$ 越接近 $S_{总}$. 因此可用 $\frac{S_{回}}{S_{总}}$ 是否趋于 1 来判断回归方程的回归效果的好坏.

9.1.4.3　复相关系数

由式(9－13)得 $1=\frac{S_{回}}{S_{总}}+\frac{S_{剩}}{S_{总}}$，即

$$\frac{S_{回}}{S_{总}}=1-\frac{S_{剩}}{S_{总}}.$$

我们定义 $R=\sqrt{1-\frac{S_{剩}}{S_{总}}}$ 为复相关系数，显然 $0\leqslant R\leqslant 1$，且 R 趋于 1 回归效果愈好.

然而在实际工程计算中，当实验样本点数较少时，计算出的 R 值一般都接近于 1，即是说用 $R\to 1$ 来检验回归效果时应注意变量个数与样本个数的适当比例. 一般认为，样本个数 n 至少应是变量个数 m 的 5～10 倍. 影响回归效果不显著还有多方面的原因，主要表现在下面几个方面：

(1) y 与 $x_1,x_2,\cdots,x_m$ 之间的关系不是线性的，而存在非线性关系，甚至 y 与 $x_1,x_2,\cdots,x_m$ 根本不存在什么关系；

(2) 影响 y 取值的，除了 $x_1,x_2,\cdots,x_m$ 以及随机误差 ε 外，还有其他不可忽略的因素；

(3) 样本的影响，比如样本数取得太少，或者样本有变异等等.

鉴于上述种种原因，我们还需对回归模型作进一步的分析.

9.1.5　线性回归模型的显著性检验

在解决实际问题时，往往事先不能判断 y 与 $x_i(i=1,2,\cdots,m)$ 之间是否确有线性关系. 因此在建模时，一般是先假定实际问题可能具有线性性，由此建立起线性回归模型；然后判断这种假设是否成立. 这就需要作假设检验.

我们可以这样考虑，如果线性回归模型能代表实际问题，即该模型是显著的，我们可以认为该模型中的参数 $\beta_0,\beta_1,\cdots,\beta_m$ 不全为零；否则认为 $\beta_0,\beta_1,\cdots,\beta_m$ 全为零. 于是按统计检验原则提出两种假设：

(1) $H_0:\beta_0,\beta_1,\cdots,\beta_m=0$；

(2) $H_1:\beta_i$ 不全为零 $(i=0,1,\cdots,m)$.

为此应用统计量

$$F=\frac{\frac{S_{回}}{m}}{\frac{S_{剩}}{(n-m-1)}}\sim F(m,n-m-1).$$

对于给定的检验水平 α，查 F 分布表可得临界值 $F_\alpha(m,n-m-1)$，并由 F 检验作如下判断：

若 $F\geqslant F_\alpha$，则表明在检验水平 α 下拒绝 H_0，从而认为线性回归模型有显著意义，因此该模型能代表实际问题，可用于工程实践；

若 $F<F_\alpha$，则表明在检验水平 α 下接受 H_0，即认为该线性回归模型不显著，不能代表

实际问题.

在例 1 中,据表 9—4 计算得

$$S_{剩}=\sum_{i=1}^{14}(y_i-\hat{y}_i)^2=36.0334,$$

$$S_{回}=\sum_{i=1}^{14}(\hat{y}_i-\bar{y})^2=3742.1396,$$

$$R=\sqrt{1-\frac{S_{剩}}{S_{总}}}=0.9810,$$

$$F=\frac{3742.1396/2}{36.0334/11}=571.1858.$$

查 F 分布表,置 $\alpha=0.01$ 得 $F_\alpha(2,11)=7.21$.

根据上述统计量,$R=0.9810$ 很接近于 1,故回归方程(9－10)的回归效果很好. 又由 $F\gg F_\alpha$ 知,线性回归方程(9－10)在检验水平 $\alpha=0.01$ 下具显著意义.

在例 2 中,对试验 D_1 进行计算:

$$\hat{y}_1=6.875,\quad \hat{y}_2=3.794,\quad \hat{y}_3=0.904,$$

$$\hat{y}_4=4.515,\quad \hat{y}_5=5.809,\quad \hat{y}_6=1.625,$$

$$S_{剩}=\sum_{i=1}^{6}(y_i-\hat{y}_i)^2=2.0146,$$

$$S_{回}=\sum_{i=1}^{6}(\hat{y}_i-\bar{y})^2=27.6953,$$

$$R=\sqrt{1-\frac{S_{剩}}{S_{总}}}=0.9655,$$

$$F=\frac{27.6953/2}{2.0146/3}=20.6209.$$

查 F 分布表,置 $\alpha=0.05$ 得 $F_\alpha(2,3)=9.55$.

根据上述统计量,$R=0.9655$ 很接近于 1,故回归方程(9－11)的回归效果很好. 又由 $F>F_\alpha$ 知,线性回归方程(9－11)在检验水平 $\alpha=0.05$ 下具显著意义.

建议读者对线性回归方程(9－12)的显著性作统计检验.

9.1.6 线性回归模型中变量显著性检验

在多元线性回归模型中,我们并不满足于线性回归模型具显著性这一结论. 因为回归方程显著并不意味着每个自变量 $x_1,x_2,\cdots,x_m$ 对因变量 y 的影响都重要. 换言之,模型中的 m 个变量有的重要,有的并不重要. 因此在模型中需保留重要变量,剔除不重要的变量,也就是精减线性回归模型.

我们可以这样理解,如果某个变量 x_i 对 y 的作用不显著,则认为它的系数 β_i 应取零值,因此检验自变量 x_i 是否重要(显著),等价于检验假设:

$H_0:\beta_i=0$;

$H_1:\beta_i\neq 0(i=1,2,\cdots,m)$.

为此引入统计量

$$F_i = \frac{\frac{b_i^2}{c_{ii}^{-1}}}{\frac{S_{剩}}{(n-m-1)}} \sim F(1, n-m-1).$$

其中 b_i 为自变量 x_i 的系数(参数估计值). c_{ii}^{-1} 为方阵 $C=(c_{ij})$, $i,j=1,2,\cdots,m$ 的逆矩阵的主对角线上的第 i 个元素. 而

$$c_{ij} = \sum_{k=1}^{n}(x_{ki}-\bar{x}_i)(x_{kj}-\bar{x}_j), \quad i,j=1,2,\cdots,m,$$

其中
$$\bar{x}_i = \frac{1}{n}\sum_{k=1}^{n} x_{ki}, \quad \bar{x}_j = \frac{1}{n}\sum_{k=1}^{n} x_{kj}.$$

对于给定的检验水平 α,查 F 分布表可得临界值 $F_\alpha(1, n-m-1)$.

若 $F_i \geqslant F_\alpha(1, n-m-1)$,则表明在检验水平 α 下拒绝假设 H_0,认为 x_i 对 y 是显著的,应留在线性回归模型中;

若 $F_i < F_\alpha(1, n-m-1)$,则表明在检验水平 α 下接受假设 H_0,认为 x_i 对 y 是不显著的,可以从线性回归模型中剔除.

一般来说,一次 F 检验只能剔除一个自变量,且这个自变量是所有不显著自变量中 F 值的最小者,在剔除一个变量后再重新建立线性回归模型并继续进行检验,直至建立的线性回归模型以及所涉及的各个自变量均呈显著性为止.

9.1.7 线性回归模型预测精度估计

回归分析的一个重要目的就是要利用回归方程进行预测. 下面分别就一元线性回归和多元线性回归两种情形进行讨论.

9.1.7.1 一元线性回归

对于一元线性回归方程

$$\hat{y} = b_0 + b_1 x,$$

所谓预测就是给定 x 的一个特定值 x_0 后,利用上述一元线性方程对 $\hat{y}$ 的相应值 $\hat{y}_0$ 进行估计. 其方法通常有两种,即点预测和区间预测.

当回归方程有意义时,用 $x=x_0$ 处的回归值 $\hat{y}_0$ 来作为 y 的估计值,这种方法即为点预测. 区间预测是当 $x=x_0$ 时,要构造两个样本函数 $\underline{\theta}$ 与 $\bar{\theta}$,使得 $P(\underline{\theta} < y_0 < \bar{\theta}) = 1-\alpha$. 此时称 $(\underline{\theta}, \bar{\theta})$ 为 y_0 的预测区间,其置信度为 $1-\alpha$.

记
$$T = \frac{y_0 - \hat{y}_0}{S^* \sqrt{1 + \frac{1}{n} + \frac{(x_0-\bar{x})^2}{S_{\bar{x}}}}}.$$

其中
$$S^* = \sqrt{\frac{S_{剩}}{(n-2)}}, \quad \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \quad S_{\bar{x}} = \sum_{i=1}^{n}(x_i - \bar{x})^2.$$

可以证明 $T \sim t(n-2)$.

给定水平 α,有
$$P\{|T| < \lambda\} = 1-\alpha,$$

即 $P\{T > \lambda\} = \frac{\alpha}{2}$,查 t 分布表可求出 λ.

由不等式 $|T| = \dfrac{|y_0 - \hat{y}_0|}{S^* \sqrt{1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{S_{\bar{x}}}}} < \lambda$，可得 y_0 的置信度为 $1-\alpha$ 的预测区间为

$$\left(y_0 - \lambda S^* \sqrt{1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{S_{\bar{x}}}},\ y_0 + \lambda S^* \sqrt{1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{S_{\bar{x}}}}\right).$$

由于 x_0 的任意性，于是 y_0 的预测区间的左边界与右边界可视为 x_0 的函数，分别为图 9－2 中所示的下曲线与上曲线. 由图 9－2 可知，当 x_0 取值愈靠近 $\bar{x}$ 时，预测区间愈窄，从而用$\hat{y}_0$ 近似 y_0 时的预测精度愈高；远离 $\bar{x}$ 时，预测精度就愈低.

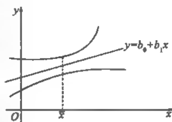


图 9－2

例 3　在硝酸钠($NaNO_3$)的溶解度试验中，测得在不同温度 x(℃)下溶解于 100 份水中的硝酸钠份数 y 的数据(见表 9－6).

表 9－6

x_i	0	4	10	15	21	29	36	51	68
y_i	66.7	71.0	76.3	80.6	85.7	92.9	99.4	113.6	125.1

设现已求得 x 与 y 的一元线性回归方程为

$$\hat{y} = b_0 + b_1 x = 67.5077 + 0.8706x,$$

试在检验水平 $\alpha = 0.05$ 下，检验所建立的回归方程是否显著，并求置信度为 95% 的 y 的预测区间以及 $x = 25$ 时 y 的预测区间.

解　本例中 $m = 1, n = 9$，据表 9－6 计算得

$$S_{剩} = \sum_{i=1}^{9} (y_i - \hat{y}_i)^2 = 6.5853,$$

$$S_{回} = \sum_{i=1}^{9} (\hat{y}_i - \bar{y})^2 = 3077.3969,$$

$$S_{总} = 3083.9822,$$

$$R = \sqrt{1 - \frac{S_{剩}}{S_{总}}} = 0.9989,$$

$$F = \frac{S_{回}/m}{S_{剩}/(n-m-1)} = \frac{3077.3967/1}{6.5853/7} = 3271.1916.$$

置 $\alpha = 0.05$，查 F 分布表得 $F_\alpha(1,7) = 5.59$.

据上述统计量，$R = 0.9989$ 很接近于 1，故回归方程的回归效果很好. 又由 $F \gg F_\alpha$，知回归方程在检验水平 $\alpha = 0.05$ 下具显著意义.

下面求预测区间：

$$S^* = \sqrt{\frac{S_{剩}}{(n-2)}} = \sqrt{6.5854/7} = 0.9699,$$

$$S_{\bar{x}} = \sum_{i=1}^{9} (x_i - \bar{x})^2 = 4060.$$

由 $P\{T > \lambda\} = \frac{0.05}{2} = 0.025$，查 $t(n-2) = t(7)$ 分布得 $\lambda = 2.3646$，故置信度为 95% 的 y 的预测区间为

$$(67.5077 + 0.8706x - 2.2934\sqrt{\frac{10}{9} + \frac{(x-26)^2}{4060}}, 67.5077 + 0.8706x + 2.2934\sqrt{\frac{10}{9} + \frac{(x-26)^2}{4060}}).$$

当 $x = 25$ 时，$y = 89.2727$ 的置信度为 95% 的预测区间为

$$(86.8550, 91.6904).$$

9.1.7.2　*多元线性回归*

对于多元线性回归方程

$$\hat{y}_i = b_0 + b_1x_1 + b_2x_2 + \cdots + b_mx_m,$$

所谓预测就是给定向量 $\boldsymbol{x} = (x_1, x_2, \cdots, x_m)^{\mathrm{T}}$ 时，用回归值来作为 y 的估计值，这个估计值称为预测值.

记

$$T = \frac{y - \hat{y}}{S^* \sqrt{1 + \frac{1}{n} + (\boldsymbol{x} - \bar{\boldsymbol{x}})^{\mathrm{T}} \boldsymbol{C}^{-1} (\boldsymbol{x} - \bar{\boldsymbol{x}})}},$$

其中

$$S^* = \sqrt{\frac{S_{剩}}{(n-m-1)}},$$

$$\bar{\boldsymbol{x}} = (\bar{x}_1, \bar{x}_2, \cdots, \bar{x}_m)^{\mathrm{T}},$$

$$\bar{x}_i = \frac{1}{n}\sum_{k=1}^{n} x_{ki},$$

$$\boldsymbol{C} = (c_{ij}), i, j = 1, 2, \cdots, m,$$

$$c_{ij} = \sum_{k=1}^{n} (x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j).$$

可以证明 $T \sim t(n-m-1)$.

给定水平 α，得置信度为 $1-\alpha$ 的 y 的预测区间为

$$(\hat{y} - \lambda S^* \sqrt{1 + \frac{1}{n} + (\boldsymbol{x} - \bar{\boldsymbol{x}})^{\mathrm{T}} \boldsymbol{C}^{-1} (\boldsymbol{x} - \bar{\boldsymbol{x}})}, \hat{y} + \lambda S^* \sqrt{1 + \frac{1}{n} + (\boldsymbol{x} - \bar{\boldsymbol{x}})^{\mathrm{T}} \boldsymbol{C}^{-1} (\boldsymbol{x} - \bar{\boldsymbol{x}})}).$$

其中 λ 由 $P(T > \lambda) = \frac{\alpha}{2}$ 查 $t(n-m-1)$ 分布表得到.

例 4　平炉炼钢过程中，由于矿石及炉气的氧化作用，铁水的总含量在不断降低，一炉钢的冶炼初期（熔化期）中总的去碳量 y 与所加的两种矿石（天然矿石与烧结矿石）的加入量 x_1、x_2 及熔化时间 x_3 有关. 经实测，某号炉的相应数据如表 9－7 所示.

(1) 求 y 关于 x_1、x_2、x_3 的线性回归方程；

(2) 检验线性回归效果是否显著（$\alpha = 0.01$）；

(3) 检验各变量 x_1、x_2、x_3 的作用是否显著（$\alpha = 0.01$）；

(4) 求 y 的置信度为 99% 的预测区间.

表 9－7

试验序号	y（吨）	x_1（槽）	x_2（槽）	x_3（5 分钟）	试验序号	y（吨）	x_1（槽）	x_2（槽）	x_3（5 分钟）
1	4.3302	2	18	50	26	2.7066	9	6	39
2	3.6485	7	9	40	27	5.6314	12	5	51
3	4.4830	5	14	46	28	5.8152	6	13	41
4	5.5468	12	3	43	29	5.1302	12	7	47
5	5.4970	1	20	64	30	4.4533	5	12	37
6	3.1125	3	12	40	31	5.3910	0	24	61
7	5.1182	3	17	64	32	4.6569	4	15	49
8	3.8759	6	5	39	33	4.5212	0	20	45
9	4.6700	7	8	37	34	4.8650	6	16	42
10	4.9536	0	23	55	35	5.3566	4	17	48
11	5.0060	3	16	60	36	4.6098	10	4	48
12	5.2701	0	18	49	37	2.3815	4	14	36
13	5.3772	8	4	50	38	3.8746	5	13	36
14	5.4849	6	14	51	39	4.591.9	9	8	51
15	4.5960	0	21	51	40	5.1588	6	13	54
16	5.6645	3	14	51	41	5.4373	5	8	100
17	6.0795	7	12	56	42	3.9960	5	11	44
18	3.2194	16	0	48	43	4.3970	8	6	63
19	5.8076	6	16	45	44	4.0622	2	13	55
20	4.7306	0	15	52	45	2.2905	7	8	50
21	4.6805	9	0	40	46	4.7115	4	10	45
22	3.1272	4	6	32	47	4.5310	10	5	40
23	2.6104	0	17	47	48	5.3637	3	17	64
24	3.7174	9	0	44	49	6.0771	4	15	72
25	3.8946	2	16	39					

解 (1) 本例中 $n = 49, m = 3$.

设 y 关于 x_1、x_2、x_3 存在线性关系

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \varepsilon.$$

将表 9－7 中的 49 组数据代入上式，得参数估计值满足的超定方程组

$$\hat{y}_i = b_0 + b_1 x_{i1} + b_2 x_{i2} + b_3 x_{i3}, \quad i = 1,2,\cdots,49,$$

其矩阵形式为

$$\boldsymbol{X}\boldsymbol{b} = \boldsymbol{y}.$$

其中

$$\boldsymbol{X} = \begin{pmatrix} 1 & x_{11} & x_{12} & x_{13} \\ 1 & x_{21} & x_{22} & x_{23} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{49\,1} & x_{49\,2} & x_{49\,3} \end{pmatrix} = \begin{pmatrix} 1 & 2 & 18 & 50 \\ 1 & 7 & 9 & 40 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 4 & 15 & 72 \end{pmatrix},$$

$$\boldsymbol{b} = (b_0, b_1, b_2, b_3)^{\mathrm{T}},$$

$$\boldsymbol{y} = (4.3302, 3.6485, \cdots, 6.0771)^{\mathrm{T}}.$$

求超定方程组 $\boldsymbol{Xb} = \boldsymbol{y}$ 的最小二乘解，对应的法方程为

$$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Xb} = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}.$$

其中

$$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} = \begin{pmatrix} n & \sum x_{i1} & \sum x_{i2} & \sum x_{i3} \\ \sum x_{i1} & \sum x_{i1}^2 & \sum x_{i1}x_{i2} & \sum x_{i1}x_{i3} \\ \sum x_{i2} & \sum x_{i2}x_{i1} & \sum x_{i2}^2 & \sum x_{i2}x_{i3} \\ \sum x_{i3} & \sum x_{i3}x_{i1} & \sum x_{i3}x_{i2} & \sum x_{i3}^2 \end{pmatrix},$$

$$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} = \left(\sum y_i, \sum x_{i1}y_i, \sum x_{i2}y_i, \sum x_{i3}y_i\right)^{\mathrm{T}}.$$

经计算，解得法方程的解向量为

$$\boldsymbol{b} = (b_0, b_1, b_2, b_3)^{\mathrm{T}} = (0.7014, 0.1604, 0.1076, 0.0359)^{\mathrm{T}}.$$

所以回归方程为

$$\hat{y} = 0.7014 + 0.1604x_1 + 0.1076x_2 + 0.0359x_3.$$

(2) 计算得

$$S_{回} = \sum_{i=1}^{49} (\hat{y}_i - \bar{y})^2 = 15.221,$$

$$S_{剩} = \sum_{i=1}^{49} (y_i - \hat{y}_i)^2 = 29.684,$$

$$F = \frac{S_{回}/m}{S_{剩}/(n-m-1)} = \frac{15.221/3}{29.684/45} = 7.6915.$$

对 $\alpha = 0.01$，查 F 分布表，得临界值 $F_\alpha(3,45) = 4.25$.

据上述统计量，因为 $F > F_\alpha$，知回归方程在检验水平 $\alpha = 0.01$ 下具显著意义.

(3) 计算 $\bar{x}_1 = 5.286, \bar{x}_2 = 11.796, \bar{x}_3 = 49.204, \bar{y} = 4.582$,

$$\widetilde{\boldsymbol{X}} = \begin{pmatrix} x_{11} - 5.286 & x_{12} - 11.796 & x_{13} - 49.204 \\ x_{21} - 5.286 & x_{22} - 11.796 & x_{23} - 49.204 \\ \vdots & \vdots & \vdots \\ x_{49\,1} - 5.286 & x_{49\,2} - 11.796 & x_{49\,3} - 49.204 \end{pmatrix},$$

$$\boldsymbol{C} = \widetilde{\boldsymbol{X}}^{\mathrm{T}}\widetilde{\boldsymbol{X}} = \begin{pmatrix} 662.000 & -918.143 & -388.857 \\ -918.143 & 1753.959 & 776.041 \\ -388.857 & 776.041 & 6247.955 \end{pmatrix}$$

$$\boldsymbol{C}^{-1} = \begin{pmatrix} 5.515 \times 10^{-3} & 2.94 \times 10^{-3} & -1.62 \times 10^{-5} \\ 2.94 \times 10^{-3} & 2.122 \times 10^{-3} & -8.52 \times 10^{-5} \\ -1.62 \times 10^{-5} & -8.52 \times 10^{-5} & 1.694 \times 10^{-4} \end{pmatrix}$$

$$F_1 = \frac{b_1^2/c_{11}^{-1}}{S_{剩}/(n-m-1)} = \frac{0.0257/5.515 \times 10^{-3}}{29.684/45} = 7.07,$$

$$F_2 = \frac{b_2^2/c_{22}^{-1}}{S_{剩}/(n-m-1)} = \frac{0.0116/2.122 \times 10^{-3}}{29.684/45} = 8.27,$$

$$F_3 = \frac{b_3^2/c_{33}^{-1}}{S_{剩}/(n-m-1)} = \frac{1.2889 \times 10^{-3}/1.694 \times 10^{-4}}{29.684/45} = 11.53.$$

当 $\alpha = 0.01$,查 F 分布表,得 $F_\alpha(1,45) = 7.24$.

因此,在检验水平 $\alpha = 0.01$ 下,$F_1 < F_\alpha$,x_1 对 y 的影响不显著;而 $F_2 > F_\alpha$,$F_3 > F_\alpha$,故 x_2、x_3 对 y 的影响是显著的.

(4) 求 y 的置信度为 99%($\alpha = 0.01$) 的预测区间:

$$S^* = \sqrt{S_{剩} / (n - m - 1)} = \sqrt{29.684/45} = 0.812.$$

由 $P(T > \lambda) = \dfrac{0.01}{2} = 0.005$,查 $t(49)$ 分布表得

$$\lambda = 2.6456.$$

于是给定水平 $\alpha = 0.01$,得置信度为 $1-\alpha$(即 99%) 的 y 的预测区间为 (t_1,t_2),其中

$$\begin{aligned} t_1 &= \hat{y} - \lambda S^* \sqrt{1 + \frac{1}{n} + (\boldsymbol{x} - \bar{\boldsymbol{x}})^{\mathrm{T}} \boldsymbol{C}^{-1} (\boldsymbol{x} - \bar{\boldsymbol{x}})} \\ &= 0.7014 + 0.1604x_1 + 0.1076x_2 + 0.0359x_3 \\ &\quad - 2.1482\sqrt{\frac{50}{49} + (\boldsymbol{x} - \bar{\boldsymbol{x}})^{\mathrm{T}} \boldsymbol{C}^{-1} (\boldsymbol{x} - \bar{\boldsymbol{x}})}, \\ t_2 &= 0.7014 + 0.1604x_1 + 0.1076x_2 + 0.0359x_3 \\ &\quad + 2.1482\sqrt{\frac{50}{49} + (\boldsymbol{x} - \bar{\boldsymbol{x}})^{\mathrm{T}} \boldsymbol{C}^{-1} (\boldsymbol{x} - \bar{\boldsymbol{x}})}. \end{aligned}$$

其中,$\bar{\boldsymbol{x}} = (\bar{x}_1,\bar{x}_2,\bar{x}_3)^{\mathrm{T}} = (5.286,11.796,49.204)^{\mathrm{T}}$,$\boldsymbol{C}^{-1}$ 如前.

只要给定 x 的一组值 $\boldsymbol{x} = (x_1,x_2,x_3)^{\mathrm{T}}$,即可计算出对应的 y 的置信度为 99% 的预测区间.

9.2 多元逐步回归分析

9.2.1 引言

多元逐步回归分析是多元线性回归分析的深入学习和研究. 设 m 个自变量取 n 组实测样本值,如表 9-8 所示.

表 9-8

y	x_1	x_2	…	x_m
y_1	x_{11}	x_{12}	…	x_{1m}
y_2	x_{21}	x_{22}	…	x_{2m}
⋮	⋮	⋮	⋮	
y_n	x_{n1}	x_{n2}	…	x_{nm}

根据上述样本数据,建立什么样的线性回归方程才能较好地反映因变量 y 与自变量 $x_1,x_2,\cdots,x_m$ 之间的统计关系呢?为此,我们要明确如下问题:

(1) 如 9.1.6 中所述,首先在回归方程 $\hat{y} = b_0 + b_1x_1 + b_2x_2 + \cdots + b_mx_m$ 中,$\hat{y}$ 并不是和所有的 x_i 都有统计意义上的关系,也许从中剔出某个变量 x_k 后更能反映对 $\hat{y}$ 的影响;其次,在回归方程中变量的重要性也不一样,通过变量的显著性检验可知,可能某些变量

对 y 更显著，而另一些变量对 y 并不显著.

(2) 什么是最优的回归方程?即从实测样本数据出发推导出的，能最好反映影响因变量 y 的回归方程. 而评价的标准可用最大复相关系数 R_{max} 或最大回归平方和 $(S_{回})_{max}$ 来衡量. 当自变量个数较少时，可用穷举法来选取最优回归方程，当自变量个数较多时采用逐步回归方法.

9.2.2 逐步回归算法的基本思路

逐步回归的基本思路是，根据自变量的重要性，每一步都选出一个重要的变量进入回归方程：

(1) 在所有可供挑选的变量中选出一个变量，使它组成的一元回归方程比其他变量有更大的回归平方和；

(2) 在剩下的自变量中选出这样一个变量，它与已选入方程的那个变量所组成的二元回归方程，比其他任一变量与已选入方程的变量所组成的二元回归方程，有更大的回归平方和；

(3) 如此继续下去，假设已经进行到第 L 步，该步是在未选入的变量中选出这样一个变量，它与已选入回归方程的变量组成 L 元回归方程，比其他余下的任何一个变量所组成的回归方程有更大的回归平方和；

(4) 考虑到较早选入回归方程的某些变量，有可能随着其后一些变量的选入而失去原有的重要性，这样的变量也应当及时从回归方程中剔除，使回归方程中始终只保留重要的变量. 例如 x_1、x_2 引入方程后，再引入 x_5，也许由于 x_5 的引入而使 x_1 的重要性下降，反而变得不重要了，此时应及时剔出 x_1.

(5) 直至最后没有变量引入，也无变量剔出时结束逐步回归算法.

综上所述，逐步回归分析的基本步骤是依次建立一系列回归方程，后一回归方程是在前一个的基础上增加或删除一个变量而构成的，增加或删除某个自变量的评判标准是用回归平方和的增加(剩余平方和的减少，复相关系数增大) 来衡量的.

9.2.3 引入自变量的依据

设已有 L 个自变量进入回归方程，即已知回归方程为

$$\hat{y} = b_0 + b_1x_1 + b_2x_2 + \cdots + b_Lx_L,$$

此时该方程相应的总离差平方和为

$$S_{总} = S_{回} + S_{剩} = U(x_1, x_2, \cdots, x_L) + Q(x_1, x_2, \cdots, x_L). \qquad (9-15)$$

现在已有的 L 个自变量所组成的回归方程中再引入一个自变量，不妨记为 $x_i (i = L+1, L+2, \cdots, m)$，于是引入 x_i 的回归方程的 $S_{总}$ 可表示为

$$S_{总} = U(x_1, x_2, \cdots, x_L, x_i) + Q(x_1, x_2, \cdots, x_L, x_i). \qquad (9-16)$$

由式(9－16) 减去式(9－15)，并注意到上述二式的总离差平方和不变，即有

$$U(x_1, x_2, \cdots, x_L, x_i) - U(x_1, x_2, \cdots, x_L) = Q(x_1, x_2, \cdots, x_L) - Q(x_1, x_2, \cdots, x_L, x_i).$$

令 $$V_i(x_1, x_2, \cdots, x_L) = U(x_1, x_2, \cdots, x_L, x_i) - U(x_1, x_2, \cdots, x_L),$$

并称 $V_i(x_1, x_2, \cdots, x_L)$ 为 x_i 对 y 的方差贡献. 即是说，若 $V_i(x_1, x_2, \cdots, x_L)$ 越大，则 x_i 对

y的影响就越大,亦即x_i对回归方程就越重要,应该引人.但是,V_i应大到什么程度自变量x_i才被引入方程呢?这就需要给出x_i的引入标准(或称引入门坎值).

统计理论表明,用统计量

$$F_{1i}=\frac{V_i(x_1,x_2,\cdots,x_L)/1}{Q(x_1,x_2,\cdots,x_L,x_i)/(n-L-2)}\sim F(1,n-L-2)$$
$$(i=L+1,L+2,\cdots,m)$$

可以检验自变量x_i是否可以引入方程.式中n是样本容量,L是已引入方程的自变量个数.

对于给定的水平值α,查F分布表可得临界值$F_\alpha(1,n-L-2)$,$F_\alpha\xlongequal{\text{记}}F_{\text{进}}$.若$F_{1i}>F_{\text{进}}$,则表明$x_i$可以引入方程;若$F_{1i}\leqslant F_{\text{进}}$,则表明$x_i$不重要,不能引入方程.需要指出,实际问题可能有多个$F_{1i}>F_{\text{进}}$,由于每次只能引入一个变量进入方程,因此,算法上我们选最大的F_{1i}所对应的变量引入,即先求

$$\max_{L<i\leqslant m}F_{1i}\xlongequal{\text{记}}F_{i_p},$$

再比较:

若$F_{i_p}>F_{\text{进}}$,则选取x_{i_p}引入;

若$F_{i_p}\leqslant F_{\text{进}}$,则表明没有变量可以引入,故引入变量的步骤到此为止.

9.2.4 剔除自变量的依据

设已有L个自变量引入回归方程,即已知回归方程为

$$\hat{y}=b_0+b_1x_1+b_2x_2+\cdots+b_Lx_L,$$

此时方程的总离差平方和可表示为

$$S_{\text{总}}=S_{\text{回}}+S_{\text{剩}}=U(x_1,x_2,\cdots,x_L)+Q(x_1,x_2,\cdots,x_L).\tag{9-17}$$

现在已有的L个自变量中剔除一个自变量,不妨剔除$x_i(i=1,2,\cdots,L)$,于是可得剔除x_i后的回归方程

$$\hat{y}=b_0'+b_1'x_1+\cdots+b_{i-1}'x_{i-1}+b_{i+1}'x_{i+1}+\cdots+b_L'x_L,\tag{9-18}$$

该方程的总离差平方和为

$$S_{\text{总}}=U(x_1,x_2,\cdots,x_{i-1},x_{i+1},\cdots,x_L)+Q(x_1,x_2,\cdots,x_{i-1},x_{i+1},\cdots,x_L),\tag{9-19}$$

由式(9－17)减去式(9－19)可得

$$U(x_1,x_2,\cdots,x_L)-U(x_1,x_2,\cdots,x_{i-1},x_{i+1},\cdots,x_L)$$
$$=Q(x_1,x_2,\cdots,x_{i-1},x_{i+1},\cdots,x_L)-Q(x_1,x_2,\cdots,x_L).$$

令　$V_i(x_1,x_2,\cdots,x_L)=U(x_1,x_2,\cdots,x_L)-U(x_1,x_2,\cdots,x_{i-1},x_{i+1},\cdots,x_L)$,

仍称V_i为x_i对y的方差贡献.若V_i越大说明x_i对y的影响越大,即x_i对y越重要,当然在方程中不能被剔除;反之V_i越小说明x_i应被剔除.然而V_i小到什么程度自变量x_i才考虑从方程中剔除呢?同前面讨论的一样,应该给出剔除x_i的门坎值.这里引入统计量F_{2i},即

$$F_{2i}=\frac{V_i(x_1,x_2,\cdots,x_L)/1}{Q(x_1,x_2,\cdots,x_{i-1},x_{i+1},\cdots,x_L)/(n-L-1)}\sim F(1,n-L-1)$$
$$(i=1,2,\cdots,L).$$

用 F_{2i} 来检验方程中哪个变量可被考虑剔除. 对于给定的水平值 α，查 F 分布表可得临界值

$$F_\alpha(1, n-L-1) \xrightarrow{\text{记}} F_{出}.$$

若 $F_{2i} \leqslant F_{出}$，则 x_i 应从方程中剔除；

若 $F_{2i} > F_{出}$，则 x_i 不应从方程中剔除.

同样需要说明的是，实际问题可能有多个 $F_{2i} \leqslant F_{出}$，由于每次只能从方程中剔除一个变量，因此在算法上我们选最小的 F_{2i} 值所对应的变量作为剔除值，即先求

$$\min_{1\leqslant i\leqslant L} F_{2i} \xrightarrow{\text{记}} F_{i_q},$$

再比较：

若 $F_{i_q} \leqslant F_{出}$，则应选取 x_{i_q} 从回归方程中剔除；

若 $F_{i_q} > F_{出}$，则方程中没有变量被剔除，转入是否可以引入新变量的讨论.

综上所述，下面给出逐步回归算法步骤：

设回归自变量为 $x_1, x_2, \cdots, x_m$.

(1) 作 m 个一元线性回归方程，取 $S_{回}$ 最大的那个方程所对应的变量入选，比如 x_{i_1} $(1 \leqslant i_1 \leqslant m)$；

(2) 对余下的 $m-1$ 个自变量，依次与 x_{i_1} 作 $m-1$ 个二元线性回归方程，仍取 $S_{回}$ 最大的那个方程所对应的变量 x_{i_2} 入选 $(1 \leqslant i_2 \leqslant m-1)$；

(3) 设已进行至第 L 步，入选的回归自变量分别为 $x_{i_1}, x_{i_2}, \cdots, x_{i_L}$.

注 在上述各步中，每一个新变量入选原则是从所属回归方程的最大的 $(S_{回})_{\max}$ 作为入选条件，但是在算法实施中，则是在每一步定出一个入选的门坎值，由此形成入选的门坎准则，用该准则判定某变量是否入选（具体见 9.2.3 引入自变量的依据）.

(4) 对于已入选的 L 个自变量，可能会出现由于新变量的引入而使原先的变量变得不重要，对于不重要的旧变量要及时剔出. 在算法中仍需定出一个剔出的门坎值和相应的剔出准则，以此判别某变量是否应被剔出（见 9.2.4 剔除自变量的依据）.

(5) 直至没有变量被引入，同时也没有变量被剔出为止，结束逐步回归算法.

上述计算步骤只是给出了逐步回归算法的一个框架，至于逐步回归算法的具体实现可以参阅本书列出的参考书目[7].

9.3 趋势面分析

9.3.1 数据变化与趋势面分析

趋势面分析是回归分析的应用，在地学等三维数据处理领域具有较为广泛的应用.

设某种观测数据 z 与另外两种数据 x, y 相关，为了分析它的变化情况，通常在较大的范围内变动 x, y，以观测 z 的取值. z 值的变化往往受到三个因素的影响，从而可以分解为以下三个部分：

(1) 由大区域因素决定的部分，它反映了在较大的区域范围内，z 随 x, y 变化的特点，亦即在大区域内数据 y 变化的一般规律.

(2) 由局部因素决定的部分，它反映了在局部区域范围内，数据 z 有异于一般规律的变化，这是一种局部的异常.

(3) 由随机因素决定的部分，它是随机性的干扰所造成的偏差.

分析 z 数据变化的三个组成部分，意在排除随机性的干扰，找出大区域内 z 值变化的一般规律和数据分布的总趋势，进而确定局部异常所在. 这样的分析在实际问题中很有意义.

从一定意义上讲，所谓趋势就是排除了局部起伏(局部异常)以后比较规则的变化. 图 9－3 中的直线代表天然气产量的总的趋势，该直线称为趋势线.

是不是任何事物在某一区域都存在趋势呢?不一定. 如图 9－4 所示，显然图 9－4 所示(a) 图趋势明显，(b) 图趋势不明显，(c) 图不存在趋势.

综上所述，趋势面分析就是提供一种定量分析方法，研究某一事物在大范围内是否存在规律性的变化，在局部范围是否存在特殊性的变化.

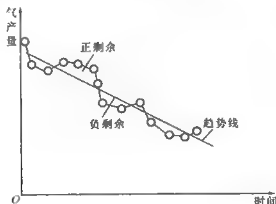


图 9－3

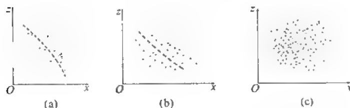


图 9－4

9.3.2 趋势面的最小二乘解(曲面拟合)

如何定量地描述事物变化的趋势呢?通常我们用曲面拟合的方法来确定三维空间的曲面或三维空间以上的超曲面以反映数据变化的趋势.

设实测数据为 $(x_i, y_i, z_i)(i=1,2,\cdots,n)$，用曲面 $z=f(x,y)$ 拟合实测数据，曲面函数的形式通常取为

$$f(x,y)=a_0\varphi_0(x,y)+a_1\varphi_1(x,y)+\cdots+a_m\varphi_m(x,y) \tag{9-20}$$

其中 $\varphi_i(x,y)(i=0,1,2,\cdots,m,m<n)$ 称为基函数，要求它们在结点 (x_i,y_i) $(i=0,1,2,\cdots,m)$ 上构成线性无关；而 $a_0,a_1,\cdots,a_m$ 则为待定的参数. 此时曲面 $f(x,y)$ 可表示为 $F(x,y,a_0,a_1,\cdots,a_m)$. 在观测点 (x_i,y_i) 上，z 数据的实测值为 z_i，趋势值为 $F(x_i,y_i,a_0,a_1,\cdots,a_m)$. 据最小二乘原理，求待定参数 $a_0,a_1,\cdots,a_m$，使以 $a_0,a_1,\cdots,a_m$ 为自变量的函数

$$Q(a_0,a_1,\cdots,a_m)=\sum_{i=1}^{n}\left[F(x_i,y_i,a_0,a_1,\cdots,a_m)-z_i\right]^2$$

达到最小值，从而 $a_0,a_1,\cdots,a_m$ 应满足方程组

$$\frac{\partial Q}{\partial a_j}=0,\quad j=0,1,\cdots,m. \tag{9-21}$$

由此可推导出法方程组(9－21)的矩阵形式：

$$\begin{bmatrix}(\boldsymbol{\varphi}_0,\boldsymbol{\varphi}_0) & (\boldsymbol{\varphi}_0,\boldsymbol{\varphi}_1) & \cdots & (\boldsymbol{\varphi}_0,\boldsymbol{\varphi}_m)\\(\boldsymbol{\varphi}_1,\boldsymbol{\varphi}_0) & (\boldsymbol{\varphi}_1,\boldsymbol{\varphi}_1) & \cdots & (\boldsymbol{\varphi}_1,\boldsymbol{\varphi}_m)\\\vdots & \vdots & & \vdots\\(\boldsymbol{\varphi}_m,\boldsymbol{\varphi}_0) & (\boldsymbol{\varphi}_m,\boldsymbol{\varphi}_1) & \cdots & (\boldsymbol{\varphi}_m,\boldsymbol{\varphi}_m)\end{bmatrix}\begin{bmatrix}a_0\\a_1\\\vdots\\a_m\end{bmatrix}=\begin{bmatrix}(\boldsymbol{\varphi}_0,\boldsymbol{z})\\(\boldsymbol{\varphi}_1,\boldsymbol{z})\\\vdots\\(\boldsymbol{\varphi}_m,\boldsymbol{z})\end{bmatrix}, \tag{9-22}$$

其中

$$\boldsymbol{\varphi}_j=[\varphi_j(x_1,y_1),\varphi_j(x_2,y_2),\cdots,\varphi_j(x_n,y_n)]^{\mathrm{T}},$$

$$(\boldsymbol{\varphi}_j,\boldsymbol{\varphi}_k)=\boldsymbol{\varphi}_j^{\mathrm{T}}\boldsymbol{\varphi}_k=\sum_{i=1}^{n}\varphi_j(x_i,y_i)\,\varphi_k(x_i,y_i),$$

$$(\boldsymbol{\varphi}_j,\boldsymbol{z})=\boldsymbol{\varphi}_j^{\mathrm{T}}\boldsymbol{z}=\sum_{i=1}^{n}\varphi_j(x_i,y_i)z_i. \tag{9-23}$$

线性方程组(9－22)的系数矩阵是一个实对称矩阵，当函数系 $\varphi_0(x,y),\varphi_1(x,y),\cdots,\varphi_m(x,y)$ 线性无关时，该方程组的解存在且唯一，记为向量 $\boldsymbol{a}^*=(a_0^*,a_1^*,\cdots,a_m^*)^{\mathrm{T}}$，并称 $\boldsymbol{a}^*$ 为函数 $Q(a_0,a_1,\cdots,a_m)$ 的最小二乘解. 拟合曲面为

$$z=F(x,y,a_0^*,a_1^*,\cdots,a_m^*)=a_0^*\varphi_0(x,y)+a_1^*\varphi_1(x,y)+\cdots+a_m^*\varphi_m(x,y), \tag{9-24}$$

z 数据的趋势值为

$$\hat{z}_i=a_0^*\varphi_0(x_i,y_i)+a_1^*\varphi_1(x_i,y_i)+\cdots+a_m^*\varphi_m(x_i,y_i). \tag{9-25}$$

9.3.3 多项式趋势面及参数估计

我们常用函数所代表的曲面来表示事物的变化趋势，而在这些函数所代表的曲面中，多项式函数所代表的曲面不仅计算方便，且当次数增高时在一定的条件下可以任意接近实测数据，并且它所代表的曲面也易于观察，所以在实际中得到广泛应用.

当曲面函数 $F(x,y,a_0,a_1,\cdots,a_m)$ 为多项式时，此时基函数 $\varphi_i(x,y)(i=0,1,\cdots,m)$ 通常可以取为 x,y 的幂函数的乘积形式. 各种类型的多项式函数的几何表示如下：

二元一次多项式 $z=a_0+a_1x+a_2y$，其几何表示为三维空间的一个平面(图 9－5).

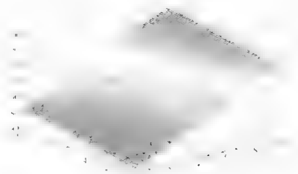


图 9－5

二元二次多项式 $z=a_0+a_1x+a_2y+a_3x^2+a_4xy+a_5y^2$，其几何表示为三维空间的一个曲面(图 9－6)，即：或为椭圆面，或为双曲面，或为抛物面.



图 9－6

二元三次多项式 $z=a_0+a_1x+a_2y+a_3x^2+a_4xy+a_5y^2+a_6x^3+a_7x^2y+a_8xy^2+a_9y^3$，其几何表示为三维空间的一个三次曲面.

一般来说，二元 p 次多项式可表示为

$$z=a_0+a_1x+a_2y+a_3x^2+a_4xy+a_5y^2+\cdots+a_my^p. \qquad (9-26)$$

式(9－26)中，p 为多项式的次数；$a_0,a_1,\cdots,a_m$ 为多项式的待定参数；m 为多项式的项数(不包括常数项)，且

$$m=\frac{(p+1)(p+2)}{2}-1.$$

例如：

$p=1,m=2$，待定参数为 a_0,a_1,a_2；

$p=2,m=3$，待定参数为 a_0,a_1,a_2,a_3；

$p=3,m=9$，待定参数为 $a_0,a_1,a_2,\cdots,a_9$；

……

所谓确定趋势面模型就是确定式(9 26)中的诸待定参数 $a_0, a_1, \cdots, a_m$.

下面介绍如何用多元线性回归的方法来估计拟合曲面中的待定参数.

给定实测数据 $(x_i, y_i, z_i)(i = 1,2,\cdots,n)$.

(1) 讨论二元一次趋势面. 设拟合平面为

$$z = a_0 + a_1 x + a_2 y, \tag{9-27}$$

在式(9-27)中,令 $x_1 = x, x_2 = y$,有

$$z = a_0 + a_1 x_1 + a_2 x_2,$$

对应的估计方程为

$$z = b_0 + b_1 x_1 + b_2 x_2. \tag{9-28}$$

这是一个二元线性回归问题,将实测数据 $x_{i1}(x_i), x_{i2}(y_i), z_i(i = 1,2,\cdots,n)$ 代人式(9-28),得超定方程组 $Xb = z$,其中

$$X = \begin{pmatrix} 1 & x_{11} & x_{12} \\ 1 & x_{21} & x_{22} \\ \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} \end{pmatrix} = \begin{pmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & y_n \end{pmatrix},$$

$$b = (b_0, b_1, b_2)^{\mathrm{T}},$$

$$z = (z_1, z_2, \cdots, z_n)^{\mathrm{T}}.$$

将 9.1.3 中关于 x_1, x_2, y 的数据表 9-2 转换成 x, y, z 的数据表 9-9.

表 9-9

x_i	y_i	$x_i y_i$	x_i^2	y_i^2	z_i	$x_i z_i$	$y_i z_i$
x_1	y_1	$x_1 y_1$	x_1^2	y_1^2	z_1	$x_1 z_1$	$y_1 z_1$
x_2	y_2	$x_2 y_2$	x_2^2	y_2^2	z_2	$x_2 z_2$	$y_2 z_2$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
x_n	y_n	$x_n y_n$	x_n^2	y_n^2	z_n	$x_n z_n$	$y_n z_n$
$\sum_{i=1}^{n} x_i$	$\sum_{i=1}^{n} y_i$	$\sum_{i=1}^{n} x_i y_i$	$\sum_{i=1}^{n} x_i^2$	$\sum_{i=1}^{n} y_i^2$	$\sum_{i=1}^{n} z_i$	$\sum_{i=1}^{n} x_i z_i$	$\sum_{i=1}^{n} y_i z_i$

由表 9-9 可分别得到法方程组的系数矩阵和右端项的元素,即

$$X^{\mathrm{T}}X = \begin{pmatrix} n & \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i y_i \\ \sum_{i=1}^{n} y_i & \sum_{i=1}^{n} x_i y_i & \sum_{i=1}^{n} y_i^2 \end{pmatrix},$$

$$X^{\mathrm{T}}z = \left(\sum_{i=1}^{n} z_i, \sum_{i=1}^{n} x_i z_i, \sum_{i=1}^{n} y_i z_i\right)^{\mathrm{T}}.$$

求解法方程组 $X^{\mathrm{T}}Xb = X^{\mathrm{T}}z$,得参数向量的估计值 $b^* = (b_0^*, b_1^*, b_2^*)^{\mathrm{T}}$,由此得趋势面方程

$$z = b_0^* + b_1^* x + b_2^* y.$$

(2) 讨论二元二次趋势面.设拟合曲面为

$$z = a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 xy + a_5 y^2. \qquad (9-29)$$

在式(9－29)中,令 $x_1 = x, x_2 = y, x_3 = x^2, x_4 = xy, x_5 = y^2$,有

$$z = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 + b_4 x_4 + b_5 x_5. \qquad (9-30)$$

这是一个五元线性回归问题,将实测数据 $x_{i1}(x_i), x_{i2}(y_i), x_{i3}(x_i^2), x_{i4}(x_i y_i), x_{i5}(y_i^2), z_i$ $(i = 1,2,\cdots,n)$ 代入式(9－30)中,得超定方程组 $\boldsymbol{Xb} = \boldsymbol{z}$,其中

$$\boldsymbol{X} = \begin{pmatrix} 1 & x_{11} & x_{12} & x_{13} & x_{14} & x_{15} \\ 1 & x_{21} & x_{22} & x_{23} & x_{24} & x_{25} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & x_{n3} & x_{n4} & x_{n5} \end{pmatrix} = \begin{pmatrix} 1 & x_1 & y_1 & x_1^2 & x_1 y_1 & y_1^2 \\ 1 & x_2 & y_2 & x_2^2 & x_2 y_2 & y_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & y_n & x_n^2 & x_n y_n & y_n^2 \end{pmatrix},$$

$$\boldsymbol{b} = (b_0, b_1, b_2, b_3, b_4, b_5)^{\mathrm{T}},$$

$$\boldsymbol{z} = (z_1, z_2, \cdots, z_n)^{\mathrm{T}}.$$

对应的法方程 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Xb} = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{z}$ 的关系矩阵和右端向量的元素可由表 9－10 列出.

表 9－10

x_i	x_1	x_2	$\cdots$	x_n	$\sum_{i=1}^{n} x_i$	z_i	z_1	z_2	$\cdots$	z_n	$\sum_{i=1}^{n} z_i$
y_i	y_1	y_2	$\cdots$	y_n	$\sum_{i=1}^{n} y_i$	$x_i z_i$	$x_1 z_1$	$x_2 z_2$	$\cdots$	$x_n z_n$	$\sum_{i=1}^{n} x_i z_i$
$x_i y_i$	$x_1 y_1$	$x_2 y_2$	$\cdots$	$x_n y_n$	$\sum_{i=1}^{n} x_i y_i$	$y_i z_i$	$y_1 z_1$	$y_2 z_2$	$\cdots$	$y_n z_n$	$\sum_{i=1}^{n} y_i z_i$
x_i^2	x_1^2	x_2^2	$\cdots$	x_n^2	$\sum_{i=1}^{n} x_i^2$	$x_i^2 z_i$	$x_1^2 z_1$	$x_2^2 z_2$	$\cdots$	$x_n^2 z_n$	$\sum_{i=1}^{n} x_i^2 z_i$
y_i^2	y_1^2	y_2^2	$\cdots$	y_n^2	$\sum_{i=1}^{n} y_i^2$	$x_i y_i z_i$	$x_1 y_1 z_1$	$x_2 y_2 z_2$	$\cdots$	$x_n y_n z_n$	$\sum_{i=1}^{n} x_i y_i z_i$
$x_i^2 y_i$	$x_1^2 y_1$	$x_2^2 y_2$	$\cdots$	$x_n^2 y_n$	$\sum_{i=1}^{n} x_i^2 y_i$	$y_i^2 z_i$	$y_1^2 z_1$	$y_2^2 z_2$	$\cdots$	$y_n^2 z_n$	$\sum_{i=1}^{n} y_i^2 z_i$
$x_i y_i^2$	$x_1 y_1^2$	$x_2 y_2^2$	$\cdots$	$x_n y_n^2$	$\sum_{i=1}^{n} x_i y_i^2$						
y_i^3	y_1^3	y_2^3	$\cdots$	y_n^3	$\sum_{i=1}^{n} y_i^3$						
$x_i^3 y_i$	$x_1^3 y_1$	$x_2^3 y_2$	$\cdots$	$x_n^3 y_n$	$\sum_{i=1}^{n} x_i^3 y_i$						
$x_i y_i^3$	$x_1 y_1^3$	$x_2 y_2^3$	$\cdots$	$x_n y_n^3$	$\sum_{i=1}^{n} x_i y_i^3$						
y_i^4	y_1^4	y_2^4	$\cdots$	y_n^4	$\sum_{i=1}^{n} y_i^4$						

由表 9 － 10 得到

$$
X^{\mathrm{T}}X=\begin{pmatrix}
n & \sum_{i=1}^{n}x_i & \sum_{i=1}^{n}y_i & \sum_{i=1}^{n}x_i^2 & \sum_{i=1}^{n}x_iy_i & \sum_{i=1}^{n}y_i^2 \\
\sum_{i=1}^{n}x_i & \sum_{i=1}^{n}x_i^2 & \sum_{i=1}^{n}x_iy_i & \sum_{i=1}^{n}x_i^3 & \sum_{i=1}^{n}x_i^2y_i & \sum_{i=1}^{n}x_iy_i^2 \\
\sum_{i=1}^{n}y_i & \sum_{i=1}^{n}x_iy_i & \sum_{i=1}^{n}y_i^2 & \sum_{i=1}^{n}x_i^2y_i & \sum_{i=1}^{n}x_iy_i^2 & \sum_{i=1}^{n}y_i^3 \\
\sum_{i=1}^{n}x_i^2 & \sum_{i=1}^{n}x_i^3 & \sum_{i=1}^{n}x_i^2y_i & \sum_{i=1}^{n}x_i^4 & \sum_{i=1}^{n}x_i^3y_i & \sum_{i=1}^{n}x_i^2y_i^2 \\
\sum_{i=1}^{n}x_iy_i & \sum_{i=1}^{n}x_i^2y_i & \sum_{i=1}^{n}x_iy_i^2 & \sum_{i=1}^{n}x_i^3y_i & \sum_{i=1}^{n}x_i^2y_i & \sum_{i=1}^{n}x_iy_i^3 \\
\sum_{i=1}^{n}y_i^2 & \sum_{i=1}^{n}x_i^2y_i & \sum_{i=1}^{n}y_i^3 & \sum_{i=1}^{n}x_i^2y_i^2 & \sum_{i=1}^{n}x_iy_i^3 & \sum_{i=1}^{n}y_i^4
\end{pmatrix},
$$

$$
A^{\mathrm{T}}\dot{z}=\left(\sum_{i=1}^{n}z_i,\sum_{i=1}^{n}x_iz_i,\sum_{i=1}^{n}y_iz_i,\sum_{i=1}^{n}x_i^2z_i,\sum_{i=1}^{n}x_iy_iz_i,\sum_{i=1}^{n}y_i^2z_i\right)^{\mathrm{T}}.
$$

解法方程组 $X^{\mathrm{T}}Xb=A^{\mathrm{T}}z$，得参数向量的估计值 $b^*=(b_0^*,b_1^*,b_2^*,b_3^*,b_4^*,b_5^*)^{\mathrm{T}}$，并由此得到趋势面方程

$$
z=b_0^*+b_1^*x+b_2^*y+b_3^*x^2+b_4^*xy+b_5^*y^2.
$$

例 5　为了分析某花岗岩体顶板形态，采集实测数据作趋势面分析. 表 9 － 11 中列出该花岗岩体共 14 个实测点的坐标值.

表 9 － 11

编号	1	2	3	4	5	6	7	8	9	10	11	12	13	14
x(km)	0	1	1	2	3	3	3	4	4	5	5	6	6	6
y(km)	0	2	5	1	2	4	7	1	3	3	6	1	4	7
z(m)	40	30	60	50	50	70	90	60	70	80	70	80	60	50

三维坐标图如图 9 － 7 所示.

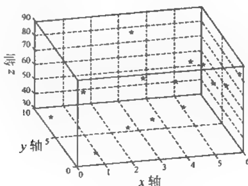


图 9 － 7

解 (1) 二元一次趋势面分析：

$$z = a_0 + a_1 x + a_2 y,$$

令 $x_1 = x, x_2 = y$，于是有

$$z = a_0 + a_1 x_1 + a_2 x_2,$$

对应的估计方程为

$$\hat{z} = b_0 + b_1 x_1 + b_2 x_2.$$

令

$$\boldsymbol{z} = (z_1, z_2, \cdots, z_{14})^{\mathrm{T}}$$

据表 9 − 11，列出法方程组 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{b} = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{z}$ 的系数矩阵和右端项如下：

$$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} = \begin{bmatrix} n & \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i y_i \\ \sum_{i=1}^{n} y_i & \sum_{i=1}^{n} x_i y_i & \sum_{i=1}^{n} y_i^2 \end{bmatrix} = \begin{bmatrix} 14 & 49 & 46 \\ 49 & 223 & 181 \\ 46 & 181 & 220 \end{bmatrix},$$

$$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{z} = \left(\sum_{i=1}^{n} z_i, \sum_{i=1}^{n} x_i z_i, \sum_{i=1}^{n} y_i z_i\right)^{\mathrm{T}} = (860, 3230, 3020)^{\mathrm{T}},$$

解得

$$\boldsymbol{b}^* = (b_1^*, b_2^*, b_3^*)^{\mathrm{T}} = (43.11, 3.59, 1.75)^{\mathrm{T}},$$

故趋势面方程为

$$\hat{z} = 43.11 + 3.59x + 1.75y.$$

该趋势平面如图 9 − 8 所示.

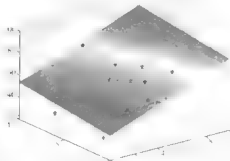


图 9 − 8

求得的 14 个观测点的趋势值和剩余值(取到个位数) 列于表 9 − 12.

表 9 − 12

编号	1	2	3	4	5	6	7	8	9	10	11	12	13	14
x(km)	0	1	1	2	3	3	3	4	4	5	5	6	6	6
y(km)	0	2	5	1	2	4	7	1	3	3	6	1	4	7
z(m)	40	30	60	50	50	70	90	60	70	80	70	80	60	50
$\hat{z}$(m)	43	50	55	52	57	60	66	59	62	66	71	66	71	76
剩余值(m)	−3	−20	−5	2	7	−10	−24	−1	−8	−14	1	−14	9	26

作趋势面等值线(图 9 －9 中平行线)和剩余等值面(图 9　9 中闭合曲线范围内)图.

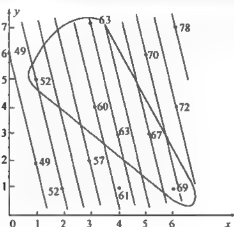


图 9 －9

(2) 二元二次趋势面分析(略).

该趋势面的效果图如图 9 － 10 所示.

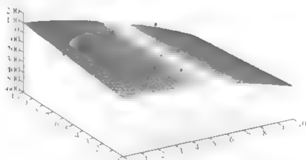


图 9 － 10

附录Ⅰ　Matlab 及其应用

1.1　Matlab 简介

1.1.1　数学软件

熟悉计算机的人会接触各式各样的软件，诸如系统软件、编译软件、应用软件、杀毒软件，等等. 上个世纪 80 年代以来，一类新的计算机软件——数学软件问世，即所谓 3M 软件.

1980 年，Wolfram 公司推出 Mathematica 软件；1982 年，Mathworks 公司推出 Matlab 软件；此后 Materloo 大学又推出一套数学软件，即 Maple 软件.

上述三种软件成为当今世界上最流行的数学软件，它们为解决现代科学计算，工程技术问题，生产经营管理问题提供了重要工具.

Mathematica 软件着重符号运算，作图功能；Matlab 软件着重数值计算，模拟功能；Maple 软件则是强大的数学软件.

1.1.2　什么是 Matlab

Matlab 的直接含义是矩阵实验室(Matrix laboratory). 最初，它是为两个矩阵软件包(Linpack，Eispack)提供接口技术，以后逐渐发展成为通用科技计算、视图交互系统和程序语言.

Matlab 语言区别于其他高级语言(如 Fortran 语言，C 语言)的最大特点是它的编辑、编译、连接和执行等四个步骤融为一体. 它能在同一画面上进行灵活操作，快速排除输入程序中的书写、语法乃至语意方面的错误，从而加快了用户编写、修改和调试程序的速度.

与其他高级语言比较，Matlab 的语法规则简单，更加贴近人的思维方式. 用 Matlab 写程序，犹如在一张演算纸上排列公式和求解问题，编程效率很高. 因此有人称它为“演算纸式”科学工程的算法语言. 下面我们举例说明 Matlab 的一些基本特点.

(1)基本数据单位是矩阵或数组，且无须定义矩阵或数组的维数，直接输入.

例 1　输入两个三维矩阵 a,b，在 Matlab 的命令空间(Command)发布命令(指令)

```
a=[1  2  3;4  5  6;7  8  9];
b=[1  4  7;8  9  10;11  12  13];
```

也可写成

```
a=[1  2  3
   4  5  6
   7  8  9];
```

```
b=[1   4   7
    8   9   10
    11  12  13];
```

注 若在上述命令中不加分号而直接键入回车换行键↙,则屏幕显示:

```
a=
   1  2  3
   4  5  6
   7  8  9
b=
    1    4    7
    8    9   10
   11   12   13
```

若加分号再键入回车换行键↙,则此时光标转入第二行.

(2)指令表达与数学习惯和工程中的常用习惯相似.

例 2 矩阵的加、减、乘.

在命令空间发布指令:

```
a=[1  2  3;4  5  6;7  8  9];
b=[1  4  7;8  9  10;11  12  13];
c=a+b↙
```

屏幕显示:

```
c=
    2    6   10
   12   14   16
   18   20   22
```

在命令空间发布指令:

```
x=[2  3  4  5;1  2  2  1];
y=[0  1  1;1  1  0;0  0  1;1  0  0];
z=x*y↙
```

屏幕显示:

```
z=
   8  5  6
   3  3  3
```

例 3 在 Matlab 中,矩阵的除法用于解线性方程组.

设 $ax=b$,其中 a 为方阵,x 为列向量,b 为列向量,则指令为

x=a\b=inv(a)*b　　(即 $a^{-1}b$)

设 $xa=b$,其中 x,b 均为行向量,则指令为

x=b/a=b*inv(a)　　(即 ba^{-1})

注 在指令 a\b 中,a 为方阵,b 为列向量(a,b 同行数). 若输入时 b 为行向量,则应

写为 a\b′.

(3)易于扩展,允许用户自定义 M 文件,从而扩大内部函数的功能.

Matlab 有大量的内部函数,可以在命令空间直接调用. 为了扩大内部函数的功能,还允许用户自己定义 M 文件(包括函数文件和文本文件). M 文件在编辑空间(Edit)编写,在命令空间调用. 我们称自定义的 M 文件为外部函数.

例 4 设

$$f(x)=\frac{1}{(x-0.3)^2+0.01}+\frac{1}{(x-0.9)^2+0.04}-6.$$

式中,$x\in[-1,2]$,取步长为 0.01,试计算 $f(x)$的值,并画出图形,再求 $f(x)$的极小点.

解 求解上述问题,可分述为如下三个步骤.

①自定义 Matlab 的外部函数,命名为 f.m.

在 Matlab 的编辑空间录入两行:

```
function y=f(x)
y=1./((x-0.3).∧2+0.01)+1./((x-0.9).∧2+0.04)-6;
```

②将上述函数文件 f.m 存盘.

有两种存盘方式,一种是将该文件存入 Matlab 的文件夹 bin 中,此时在存盘对话框中选取"m 文件格式",以后调用该文件时,在命令空间直接发布命令:

```
f↙
```

另一种存盘方式是将该文件存入其他文件夹中,此时在存盘对话框中选取"所有文件格式",以后调用该文件时要首先确定路径. 例如,若将文件 f.m 存入 C 盘中的文件夹 My Documents 中,即

```
C:\My Documents\f.m
```

要在命令空间调用 f.m,则需要 Matlab 的内部函数 path 显示当前 Matlab 的搜索路径,即在命令空间发布如下指令:

```
path('C:\My Documents',path)↙
f↙
```

③存盘文件 f.m 后退出编辑空间,进入 Matlab 的命令空间,发布如下指令:

```
x=-1:0.01:2;
y=f(x)↙
```

屏幕显示出 $f(x)$的 300 个函数值.

若发布如下指令:

```
x=-1:0.01:2;
plot(x,f(x))↙
```

屏幕显示出 $f(x)$的图形(见附录图 1).

若发布指令:

```
xm=fmin('f',0.5,1)↙
```

屏幕显示 $f(x)$在 0.5~1 之间的一个极小点,即

xm=

0.6370

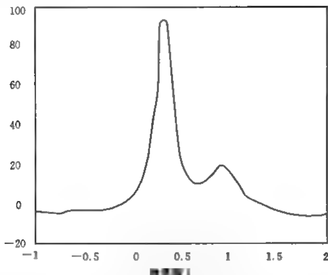
当求其极小值时，可发布指令：

y=f(xm)↙

屏幕显示 $f(x)$在极小点 xm 处的极小值：

y=

11.2528



上面我们用了三个主程序分别完成例 4 的三项要求（计值、画图、求极小点），其实我们可以用一个文本文件来完成这三项要求. 首先在编辑空间写入文本文件 L.m，它包含如下指令：

```
x=-1:0.01:2;
y=f(x);
plot(x,f(x));
xm=fmin('f',0.5,1)
```

将文件 L.m 在编辑空间存盘，然后在命令空间运行，即

L.m↙

屏幕将陆续显示 $f(x)$的值、图形和一个极小点"xm".

在上述指令中，$f(x)$是外部函数（自定义），它必须在编辑空间中编写存盘后方可调用；而 plot，fmin 等是内部函数，它可在命令空间直接调用.

(4)Matlab 易学易用.

在前面的四个例子中，以下指令：

c=a+b　　z=x*y　　x=a\b　xm=fmin('f',0.5,1)

与数学习惯十分相似，易学易用. 而这些指令若用其他高级语言来编写则相当困难，它们

需用 Fortran 或 C 语言的几十条或几百条语句才能完成.

1.1.3　Matlab 的主要用途

Matlab 软件的主要用途如下：

(1)作为高等学校有关课程的基本教学工具. 例如数学实验、数学建模、实用线性代数、数值分析、最优化方法、最优控制、数理统计、数字信息处理、时间序列分析及动态系统仿真等课程均适宜使用 Matlab 作为教学工具. Matlab 软件是大学本科生、硕士生、博士生必须掌握的基本技能.

(2)在工程领域,Matlab 提供的工具箱以及由 M 文件扩展的工具箱成为工程师进行设计、科研的强大工具. Matlab 由主包(含几百个核心的内部函数)和 30 多个工具箱(Toolbox)构成,而 Toolbox 可分为两类：

功能性工具箱　扩充 Matlab 的符号计算功能、图视建模仿真功能、文字处理功能、硬件实时互换功能.

学科性工具箱　各种专业工具,例如,Optimal Toolbox,Control Toolbox,Communication Toolbox,Signal Toolbox,等等.

1.1.4　几点说明

(1)在 Matlab 语言中,英文字母大小写要加以区别,如 A 与 a 代表不同的变量.

(2)所有的内部函数、外部函数名均由小写字母构成,如 inv(a)有意义,但 INV(a)无意义.

(3)标识符(变量,函数)以字母打头,最多 19 个字符.

(4)help(内部命令)提供在线帮助,如调用 helpfmin 将显示内部函数 fmin 的内容.

(5)demo(内部命令)提供功能演示,在 Matlab 的 Command 空间中有一个菜单项 Help,在其下拉菜单中有一个子菜单 Examples and Demos,其中列出了最优化方法的各种算例.

(6)path(内部命令)显示当前 Matlab 搜索路径.

1.1.5　矩阵运算

Matlab 在进行矩阵运算方面,显得特别简捷、高效和方便.

例 5　矩阵函数运算：

输入矩阵

```
a=[1.5  1.1  1.3;  2.1  2.3  2.5;  3.3  3.5  3.1]↙
```

屏幕显示：

```
a=
    1.5000  1.1000  1.3000
    2.1000  2.3000  2.5000
    3.3000  3.5000  3.1000
```

求 a 的转置. 发布指令：

a'↙

ans =

1.500 0　2.100 0　3.300 0

1.100 0　2.300 0　3.500 0

1.300 0　2.500 0　3.100 0

注　第一个指令中输入矩阵指定了赋值变量 a，其后未加分号“;”，故键入“↙”后直接显示变量名 a 的结果；第 2 个指令中，转置矩阵未指定赋值变量，故键入“↙”后显示变量名“ans”的结果.

求 a 的逆. 发布指令：

inv(a)↙

ans=

1.9565　−1.3766　0.2899

−2.1014　−0.4348　1.2319

0.2899　1.9565　1.3768

求 a 的特征值. 发布指令：

eig(a)↙

ans=

6.9000

0.3464

−0.3464

对 a 作 LU 分解. 发布指令：

[L,U]=lu(a)↙

L=

0.4545　1.000　0

0.6364　−0.1481　1.000

1.000　0　0

U=

3.3000　3.5000　3.1000

0　−0.4909　−1.1091

0　0　0.5111

关于矩阵函数运算的内容还很多，如对矩阵进行正交变换、奇异值分解、求矩阵的特征值和特征向量、求矩阵的秩和条件数以及建立矩阵函数，等等.

例 6　矩阵的截取与合成.

输入矩阵　a=[1　2　3;4　5　6;7　8　9;10　11　12];

指令 a(i,:)表示取第 i 行所有的元素；a(:,i)表示取 a 的第 i 列所有的元素. 例如发布指令：

a(4,:)↙　　　(取 a 的第 4 行)

```
ans=
     10  11  12
a(1:3,:)↙          (取a的1~3行)
ans=
     1  2  3
     4  5  6
     7  8  9
a(:,1:2)↙          (取a的1~2列)
ans=
      1   2
      4   5
      7   8
     10  11
a(:,3)↙            (取第3列)
ans=
      3
      6
      9
     12
```

综上,a(i,:)表示取第i行所有的元素,a(:,j)表示取第j列所有的元素,a(i:j,:)表示取第i行至第j行所有的元素,a(:,i:j)表示取第i列至第j列所有的元素.

设a是例6中的输入矩阵,再输入r,即

```
r=[13  14  15];
```

发布指令:

```
[a;r]↙
```

即将r添入到a的最后一行合成新的矩阵,屏幕显示:

```
ans=
      1   2   3
      4   5   6
      7   8   9
     10  11  12
     13  14  15
```

也可以指定赋值变量b,即

```
b=[a;[13  14  15]]↙
```

```
b=
    1   2   3
    4   5   6
    7   8   9
   10  11  12
   13  14  15
```

例 7 矩阵的除法(左除"\"与右除"/").

设 A 是 $m\times m$ 阶方阵,B 是 $m\times n$ 阶矩阵且 $\det(A)\neq 0$,其命令为

A\B ⟹ A * Z−B　　　　　　　　(Z 是 $m\times n$ 阶矩阵,等价于解矩阵方程)

⟹ A * Z(:,j)=B(:,j)　　　　　　($j=1,\cdots,n$;等价于解 n 个线性方程组)

⟹ Z(:,j)=A\B(:,j)=inv(A) * B(:,j)

($j=1,\cdots,n$;求得 n 个方程组的解)

A/B=(A\B)'

例 8 矩阵的除法(点左除".\"与点右除"./").

设 A,B 均为 $m\times n$ 阶的矩阵,"A.\B"表示B 的所有元素分别被 A 的所有对应元素相除后所形成的 $m\times n$ 阶矩阵;"A./B"表示 A 的所有元素分别被 B 的所有对应元素相除后所形成的 $m\times n$ 阶矩阵.

输入以下指令:

```
A=[1  2  3];
B=[4  5  6];
C=A.\B↙
```

屏幕显示:

```
C=
   4.0000   2.5000   2.0000
D=A./B↙
```

屏幕显示:

```
D=
   0.2500   0.4000   0.5000
```

例 9 特殊矩阵(友矩阵、零矩阵、常值矩阵).

友矩阵(compan):

设三次多项式为 x^3-7x+6,欲求该多项式的零点,即求 $x^3-7x+6=0$ 的三个根.

解 输入

```
p=[1  0  -7  6];
a=compan(p)↙
```

屏幕显示:

```
a=
    0  7  -6
    1  0   0
    0  1   0
```

再输入

```
eig(a)↙
ans=
     -3.0000
      2.0000
      1.0000
```

友矩阵 a 的三个特征值即为所给三次多项式的三个零点.

零矩阵(zeros):

zeros(i,j)表示产生 $i \times j$ 阶的零矩阵;zeros(size(a))表示产生与矩阵 a 同阶的零矩阵.

常数矩阵(ones):

ones(i,j)表示产生 $i \times j$ 阶的元素全为 1 的矩阵;ones(size(a))表示产生与 a 同阶的元素全为 1 的矩阵.

1.2 最优化方法计算

1.2.1 无约束极值算例

1)一元函数求极小点

内部命令 fmin('f',a,b,) 表示求一元函数 $f(x)$ 在区间 $[a,b]$ 内的极小点,其中 f 是在 Edit 空间编辑的一个外部函数(M 文件).

内部命令 fmin('f',x) 表示求 $f(x)$ 在初始点 x 邻近的极小点.

例 1 $f(x)=(x-3)\sqrt{x}$, $x_0=0.5$.

解 (1)在 Edit 空间编辑一个外部函数,取名"fun.m".

```
function f=fun(x)
f=(x-3)*sqrt(x);
```

(2)将 M 文件 fun 存盘.

(3)在Command 空间调用 fun.

```
x0=0.5;
x=fmin(fun,x0)↙
x=
   1.0000            (f(x)在 x=0.5 附近的极小点)
fun(x)↙
ans=
     -2.0000      (f(x)在极小点所对应的函数值)
```

2)多元函数求极值

内部命令 fminu('f',x)　表示求多元函数 $f(x)$ 在向量 $\boldsymbol{x}$ 邻近的局部极小点.

例 2　香蕉函数(Banana Function).

设 $f(x)=100(x_2-x_1^2)^2+(1-x_1)^2$,其图形如附录图 2 所示. $\boldsymbol{x}^{(0)}=(-1.9,2)^T$ 为初始点,$\boldsymbol{x}^*=(1,1)^T$ 为极小点,$f(x^*)=0$.



附录图 2

演示程序　在 Matlab 的 Command 空间中,选择菜单项 Help 的子菜单 Demos,即列出求解该香蕉函数的六种算法的演示图形. 这些无约束优化算法分别是:Steepest 算法,Simplex 算法,DFP 算法,BFGS 算法,Gauss－Newton 算法,Levenbrg-Marguardt 算法(简称 L－M 算法).

解　(1)自定义外部函数,建立 M 文件.

```
function y=fun1(x)
y=100*(x(2)-x(1)∧2)∧2+(1-x(1))∧2;
```

(2)存盘 fun1.m.

(3) 调用 fun1:

```
x=[-1.9,2];
x=fminu('fun1',x)↙
```

屏幕显示:

```
x=
   1.0000     0.9999
fun1(x)↙
ans=
     1.0e-06
```

事实上,Matlab 在演示中提到的六种无约束算法其效果是不一样的. 通过演示程序可以看出:从初始点 $\boldsymbol{x}^{(0)}$ 出发能全程达到极小点 $\boldsymbol{x}^*$ 的算法有单纯形法(Simplex)、DFP 算法、BFGS 算法以及 L－M 算法;从 $\boldsymbol{x}^{(0)}$ 出发只能搜索到中途的算法是最快速下降法(Steepest);只能从 $\boldsymbol{x}^*$ 的邻近点出发而达到 $\boldsymbol{x}^*$ 的算法是 Gauss－Newton 算法(G－N);而从 $\boldsymbol{x}^{(0)}$ 出发最快速达到 $\boldsymbol{x}^*$ 的算法有 DFP,BFGS 算法.

1.2.2 约束极值

1)线性规划算法

$$\text{(LP)} \quad \min C^{T}x \quad \text{s.t.} \begin{cases} Ax \leqslant b, \\ x \geqslant 0. \end{cases}$$

其中$A_{m\times n}$,$x\in R^n$,$C\in R^n$,$b\in R^m$.

内部命令 linprog('f',A,b) 表示求上述线性规划的最优解,其中 f 为线性规划的目标函数

$$f=C^{T}x=C_1x_1+C_2x_2+\cdots+C_nx_n.$$

例 3

$$\min(2x-3y),$$
$$\text{s.t.} \begin{cases} x+y<10, \\ -x+y<-2, \\ x>1(-x<-1). \end{cases}$$

解 (1)自定义外部函数,建立 M 文件 fun2.m(用 x 表示二维向量,其分量 x(1),x(2)分别表示题中的 x,y),即

```
function f=fun2(x)
f=2*x(1)-3*x(2);
```

(2)存盘 fun2.m.

(3)调用:

```
A=[1  1;-1  1;-1  0];
b=[10  -2  -1];
x=lp('fun2',A,b)↙
x=
   6.0000     4.0000
   fun2(x)↙
ans=
       -24.0000
```

2)非线性规划算法(基本算法)

$$\text{(NLP)} \quad \begin{cases} \min f(x), \\ \text{s.t.} \quad g(x)\leqslant 0. \end{cases}$$

其中,$f(x)$为 n 元非线性函数,$g(x)$为向量值函数,$g(x)=(g_1(x),\cdots,g_m(x))^T$,要求$f(x)$与$g_i(x)(i=1,m)$其中至少有一个为非线性函数.

内部命令 constr('fg',x) f 代表目标函数,g 代表约束函数,x 代表初始点.

例 4 $\min f(x)=e^{x_1}(4x_1^2+2x_2^2+4x_1x_2+2x_2+1)$

$$\text{s.t.} \begin{cases} 1.5+x_1x_2-x_1-x_2\leqslant 0, \\ -x_1x_2-10\leqslant 0. \end{cases}$$

解 (1)自定义外部函数 fun3.

```
function[f,g]=fun3(x)
f=exp(x(1)) * (4 * x(1)∧2+2 * x(2)∧2+4 * x(1) * x(2)+2 * x(2)…+
   1);
g(1)=1.5+x(1) * x(2)-x(1) -x(2);
g(2)=-x(1) * x(2)-10;
```

(2)存盘 fun3. m.

(3)调用:

```
x=[-1  1];
x=constr('fun3'x)↙
x=
   -9.5474    1.0474
[f,g]=fun3(x)↙
f=
   0.0236
g=
    1.0e-0.14 * 0.1110-0.1776
```

3)非线性规划算法(带有参数的情形)

例 5 下界条件与上界条件.

在例 4 中,若对决策变量 x 限定在某一区域内,如 $vlb<x<vub$ 其中 vlb,vub 均为二维向量,则可用如下语句:

```
x=constr('fun3',x,options,vlb,vub);
```

又如在例 4 中,要求 $x>0$,则调用时应发布如下指令:

```
x=[-1  1];              (赋初值)
options=[      ];       (参数说明向量使用缺省选项)
vlb=[0  0];             (赋下界)
vub=[      ];           (无上界)
x=constr('fun3',x,options,vlb,vub)↙
```

经 7 次迭代后,问题的解为

```
x=0     1.5000
[f,g]=fun3(x)↙
f=
   8.5000
g-
     0
    -10
```

若决策变量 x 的一部分分量有上、下界的限制,则可通过对 vlb,vub 向量部分赋值来实现,当 vlb,vub 的分量数目 t 比向量 x 的分量数目 n 少时,则 x 中只有前 t 个分量被限定

有界.

例 6 不等式、等式混合约束.

在例 4 中,若再加入等式约束 $x_1+x_2=0$,则相应的程序变动如下.

解 (1)自定义外部函数.

```
function[f,g]=fun4(x)
f=exp(x(1))*(4*x(1)∧2+2*x(2)∧2+4*x(1)*x(2)+2*x(2)…+1);
g(1)=x(1)+x(2);
g(2)=1.5+x(1)*x(2)-x(1)-x(2);
g(3)=-x(1)*x(2)-10;
```

(2)存盘 fun4. m.

(3)调用:

```
x=[-1  1];
options(13)=1;
x=constr('fun4',x,options)↙
```

经 11 次迭代后得解:

```
x=-1.2247      1.2247
[f, g]=fun3(x)↙
f=
   1.8951
g=
      0.0000
     -0.0000
     -8.5000
```

注 参数说明向量 options 共有 18 个分量,包含了在优化程序中需要用到的所有参数.第 13 个分量的功能是等式约束条件.例如 options(13)=1 表明第一个约束条件是等式约束,options(13)=3 表明第 1 至第 3 个约束条件是等式约束,因此 1,3 表示等式约束的个数.等式约束条件必须放置在变量 g 的前几个分量中.

1.2.3 线性最小二乘问题

众所周知,已知实测数据$(t_i,y_i),i=1,2,\cdots,m$,其中 $t_i\neq t_j(i\neq j),y_i=f(t_i),i=1,2,\cdots,m$,若用多项式

$$y=\sum_{j=0}^{n}a_jt^j,\quad n\ll m$$

去拟合曲线 $y=f(t)$,则问题归结为求超定方程组的解,即解方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$,其中

$$\boldsymbol{A}=\begin{bmatrix}1 & t_1 & t_1^{\ 2} & \cdots & t_1^{\ n}\\ 1 & t_2 & t_2^{\ 2} & \cdots & t_2^{\ n}\\ \vdots & \vdots & \vdots & & \vdots\\ 1 & t_m & t_m^{\ 2} & \cdots & t_m^{\ n}\end{bmatrix},\quad \boldsymbol{x}=\begin{bmatrix}a_0\\ a_1\\ \vdots\\ a_n\end{bmatrix},\quad \boldsymbol{b}=\begin{bmatrix}y_1\\ y_2\\ \vdots\\ y_m\end{bmatrix}.$$

可以证明 $Ax=b$ 的非常义解(最小二乘解)即是如下线性最小二乘问题的解：

$$\min_{x\in R^{n+1}} \|Ax-b\|_2^2.$$

Matlab 给出了非负的线性最小二乘问题的解法：

$$\begin{cases}\min\limits_{x\in R^{n+1}} \|Ax-b\|_2^2,\\ \text{s. t. } x\geqslant 0.\end{cases}$$

Matlab 的内部函数为 lsqnonneg(A, b')，其中 A 表示超定方程的系数矩阵，b 表示右端项.

例 7 解超定方程组

$$\begin{cases}2x+4y=11,\\ 3x-5y=3,\\ x+2y=6,\\ 2x+y=7.\end{cases}$$

上述方程组的矩阵形式为 $A\bar{x}=b$，其中

$$A=\begin{bmatrix}2 & 4\\ 3 & -5\\ 1 & 2\\ 2 & 1\end{bmatrix},\quad \bar{x}=\begin{bmatrix}x\\ y\end{bmatrix},\quad b=(11,3,6,7)^{\mathrm{T}}.$$

解 直接对 Matlab 的 Command 空间发布指令：

```
A=[2  4;3  -5;1  2;2  1];
b=[11  3  6  7];
x1=lsqnonneg(A,b')↙
x1=
    3.0403
    1.2418
```

例 8 已知实测数据如下：

x_i	−1.00	−0.75	−0.50	−0.25	0	0.25	0.50	0.75	1.00
y_i	0.220 9	0.329 5	0.882 6	1.439 2	2.000 3	2.564 5	3.133 4	3.706 1	4.283 6

试用一次、二次多项式拟合 $y=f(x)$.

解 (1)一次多项式拟合. 设

$$y=a+bx,$$

并将实测数据 $(x_i,y_i)_{i=\overline{1,9}}$ 代入上式得超定方程组

$$A\bar{x}=\bar{y}.$$

其中：

$$A=\begin{bmatrix}1 & -1.0000 \\ 1 & 0.7500 \\ 1 & -0.5000 \\ \vdots & \vdots \\ 1 & 1.0000\end{bmatrix}_{9\times 2}, \bar{x}=\begin{bmatrix}a \\ b\end{bmatrix}, \quad \bar{y}=\begin{bmatrix}0.2209 \\ 0.3295 \\ 0.8826 \\ \vdots \\ 4.2836\end{bmatrix}_{9\times 1}.$$

Matlab 指令:

```
A=[1  -1.000 0;1  -0.750 0;1  -0.500 0;1  -0.250 0;1  0.000 0;…
   1  0.250 0;1  0.500 0;1  0.750 0;1  1.000];
y1=[0.220 9  0.329 5  0.882 6  1.439 2  2.000 3  2.564 5…
   3.133 4  3.706 1  4.283 6];
x1=lsqnonneg(A,y1')↙
x1=
   2.062 2
   2.133 8
```

(2)二次多项式拟合. 设

$$y=a_0+a_1x+a_2x^2,$$

将实测数据$(x_i,y_i)_{i=\overline{1,9}}$代入上式得超定方程组

$$A\bar{x}=\bar{y}.$$

其中

$$A=\begin{bmatrix}1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_9 & x_9^2\end{bmatrix}, \quad \bar{x}=\begin{bmatrix}a_0 \\ a_1 \\ a_2\end{bmatrix}, \quad \bar{y}=\begin{bmatrix}y_1 \\ y_2 \\ \vdots \\ y_9\end{bmatrix}.$$

Matlab 指令:

```
A=[1  -1.0000  1.0000;  1  -0.7500  0.5626;…;1  -0.5000  0.2500;…
   1  -0.2500  0.0625;  1  0.0000  0.0000;  1  0.2500  0.0625;…
   1  0.5000  0.2500;  1  0.7500  0.5625;  1  1.0000  1.0000];
y1=[0.2209  0.3295  0.8826  1.4392  2.0003  2.5645…
   3.1334  3.7061  4.2836];
x1=lsqnonneg(A,y1')↙
x1=
   1.9599
   2.1338
   0.2455
```

注 对于三次多项式拟合,使用 Matlab 命令将导致失败. 因为三次多项式拟合时,其系数向量$(a_0,a_1,a_2,a_3)^T$ (1.959 9,2.427 2,0.245 5, 0.388 1)^T 中的第 4 个分量为负数,这就违反了非负的条件. 在本附录 3.3 中我们将介绍多项式拟合的一般方法(去掉 x

非负的限制).

1.2.4 非线性最小二乘问题

数学模型：$\min \| F(x) \|_2^2$，其中

$$F(x)=(f_1(x),f_2(x),\cdots,f_m(x))^T, \quad x=(x_1,x_2,\cdots,x_n)^T.$$

通常将其写成平方和的极小形式

$$\min S(x) = \sum_{i=1}^{m} f_i^2(x).$$

上述问题的实际意义为：给定实测数据$(t_i,y_i)_{i=\overline{1,m}}$，其中 $t_i \neq t_j (i \neq j)$，$y_i=f(t_i)$. 如果拟合函数不是代数多项式形式，如：

$y(t,x)=x_1+x_2\exp(tx_3)$ （一元三参数问题，t 为实数，x 为三维向量），

$y(t,x)=\dfrac{x_1x_3t_1}{1+x_1t_1+x_2t_2}$ （二元三参数问题，t 为二维向量，x 为三维向量），

则采用非线性最小二乘法作偏差平方和函数，即

$$S(x) = \sum_{i=1}^{m}[y(t_i,x)-y_i]^2 = \sum_{i=1}^{m} f_i^2(x).$$

其中偏差为 $y(t_i,x)-y_i=f_i(x), \quad i=\overline{1,m}.$

Matlab 函数 lsqnonlin('f',x)给出了求解上述非线性最小二乘问题的算法，其中"f"代表平方和函数 $S(x)$，"x"是给定的初始点.

例 9 一元三参数问题. 设已知数据如下：

t_i	$t_1=-5$	$t_2=-3$	$t_3=1$	$t_4=3$	$t_5=5$	$t_6=-1$
y_i	$y_1=127$	$y_2=151$	$y_3=421$	$y_4=460$	$y_5=426$	$y_6=379$

拟合函数 $y(t,x)=x_1+x_2\exp(tx_3)$，

$$S(x) = \sum_{i=1}^{6}[y(t_i,x)-y_i]^2 = \sum_{i=1}^{6}[x_1+x_2\exp(t_ix_3)-y_i]^2.$$

给定 $x^{(0)}=(580,-180,0.16)^T, \quad S^{(0)}=0.3\times10^5.$

解 (1)自定义外部函数.

```
function S=f(x)
t=[-5  -3  1  3  5  -1];
y=[127  151  421  460  426  379];
S=0.0;
for i=1:6
S=S+(x(1)+x(2)*exp(t(i)*x(3))-y(i))^2
end
```

(2)存盘 f.m.

(3)调用：

```
x=[580  -180  -0.16];
```

```
x=lsqnonlin('f',x)↙
x=
   523.3000  -157.0000  -0.1994
S=f(x)↙
S=
   0.1300 * 10^5
```

1.3 数据分析

1.3.1 数据的输入和输出

1)键盘输入单个参数(input 命令)

例 1 $f(x)=[(x-0.3)^2+0.01]+[(x-0.9)^2+0.04]-6$，

计算 $f(x)$ 的值，其中 x 是待输入参数.

解 (1)建立函数文件 f.m(存盘).

```
function y=f(x)
y=((x-0.3)∧2+0.01)+((x-0.9)∧2+0.04)-6;
```

(2)建立文本文件 s.m(存盘).

```
x=input('inputx:');
y=f(x)
```

(3)运行 s.m.

```
s↙(在 Command 空间发布)
```

屏幕显示：

```
input:
5↙  (Command 空间发布)
```

屏幕显示：

```
y=
   32.95
```

2)用矩阵输入(小型矩阵)

例 2 解线性方程组 $Ax=b$，其中

$$A=\begin{bmatrix}1 & 1/2 & 1/3\\ 1/2 & 1/3 & 1/4\\ 1/3 & 1/4 & 1/5\end{bmatrix},\quad x=(x_1,x_2,x_3)^T,\quad b=(11/16,13/2,47/60)^T.$$

解 (1)建立文本文件 solution.m(存盘).

```
a=[1  1/2  1/3;1/2  1/3  1/4;1/3  1/4  1/5];
b=[11/16  13/12  47/60];
x=a\b'
```

(2)运行 solution.m.

```
solution ↙(在 Command 空间发布)
```

屏幕显示：

x=

1

1

1

3)用数据文件输入(大型矩阵)

例 3 经济(longley)数据.

某国(或某地区)1947 年—1962 年(共 16 年)的经济数据如附表 1－1 所示.

附表 1－1

年份	GNP	GDP	失业率	军队人数	人口	就业率
1947	×	×	×	×	×	×
1948	×	×	×	×	×	×
1949	×	×	×	×	×	×
⋮	⋮	⋮	⋮	⋮	⋮	⋮
1962	×	×	×	×	×	×

注 GNP 表示紧缩率，GDP 表示国民生产总值.

附表 1－1 内的数据构成 16×7 阶矩阵，若用前面两种方式输入显然不宜，下面用数据文件输入.

解 (1)建立数据文件(或文本文件) longley. dat，在 Edit 环境下写入如下数据块(16 行，7 列)：

× × × × × × ×

⋮ ⋮ ⋮ ⋮ ⋮ ⋮ ⋮

× × × × × × ×

(2)存盘 longley. dat.

(3)调用：

longley ↙

屏幕显示 16×7 数据块，再调用：

[n,p]=size(longley)↙

屏幕显示：

n=

16

p=

7

(4)也可建立文本文件 longley. m. 设赋值变量名为 idata，在 Command 空间键入

idata=[× × × × × × ×;… … …× × × × × × ×];

[n,p]=size(idata)

建立文本文件 longley. m 存盘后再调用：

longley. m↙

屏幕分别显示出 16×7 阶的以变量 idata 标识的数据块以及与上面相同的 n,p 值.

1.3.2 列数据分析

下列内部函数具有列数据的分析功能：

给定 longley 16×7 阶矩阵

函数	说明
max(longley)	longley 的各列的最大值所构成的行向量
min(longley)	longley 的各列的最小值所构成的行向量
mean(longley)	longley 的各列的平均值所构成的行向量
mediam(longley)	longley 的各列的中值所构成的行向量
std(longley)	longley 的各列的标准差(均方差)所构成的行向量
sum(longley)	longley 的各列元素总和所构成的行向量
prod(longley)	longley 的各列元素乘积所构成的行向量

若给定 x 为列向量，则 max(x), min(x), …, prod(x)均为数；若 x 为行向量，则它们仍为行向量.

例 4 求矩阵 longley 的误差矩阵.

一个给定矩阵的误差矩阵是由该矩阵每一列元素分别减去该列元素的均值所形成的.

解 发布如下指令：

[n,p]=size(longley)↙

屏幕显示：

```
n=
   16
p=
   7
```

e=ones(n,1)↙

屏幕显示：

$$e=\begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix}$$

m=mean(longley)↙

屏幕显示：

```
m=
   1.0e+003
   1.9550   1.0120   0.3975   0.3351   0.2798   0.1174   0.0660
```

mean=longley−e * m

屏幕显示 longley 的误差矩阵(16×7 阶).

注 设 $\boldsymbol{m}=[m(1)\quad m(2)\quad m(3)\quad m(4)\quad m(5)\quad m(6)\quad m(7)]$.

$$
\boldsymbol{e}.\boldsymbol{m}=\begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix}[m(1)\quad m(2)\cdots m(7)]
$$

$$
=\begin{bmatrix}\mathrm{m}(1) & \mathrm{m}(2) & \cdots & \mathrm{m}(7)\\ \mathrm{m}(1) & \mathrm{m}(2) & \cdots & \mathrm{m}(7)\\ \vdots & \vdots & & \vdots\\ \mathrm{m}(1) & \mathrm{m}(2) & \cdots & \mathrm{m}(7)\end{bmatrix}.
$$

longley−e * m 是从 longley 的各列分别减去各列的均值 m(i)($i=\overline{1,7}$)而形成的误差矩阵,即

$$
\begin{bmatrix}\times-\mathrm{m}(1) & \times-\mathrm{m}(2) & \cdots & \times-\mathrm{m}(7)\\ \times-\mathrm{m}(1) & \times-\mathrm{m}(2) & \cdots & \times-\mathrm{m}(7)\\ \vdots & \vdots & & \vdots\\ \times-\mathrm{m}(1) & \times-\mathrm{m}(2) & \cdots & \times-\mathrm{m}(7)\end{bmatrix}.
$$

1.3.3 实测数据归一化(标准化)

当试图对某组数据进行拟合或回归时,首先应对数据标准化,以改善结果的精确度.

设数据块 longley 为 16×7 阶矩阵.标准化该数据的方法为:

(1)移去均值,得误差矩阵.

```
e=ones(length(longley),1);
x=longley−e * mean(longley);
```

(2)标准化,即将 x 化为单位标准差矩阵.

```
std(x);
e * std(x);
x=x./(e * std(x));
```

注 设 $\mathrm{std}(x)=[S_1,S_2,\cdots,S_7]$,则

$$
\mathrm{e.std}(x)=\begin{bmatrix}S_1, & S_2, & \cdots & S_7\\ S_1, & S_2, & \cdots & S_7\\ \vdots & \vdots & & \vdots\\ S_1, & S_2, & \cdots & S_7\end{bmatrix}.
$$

(3)对单位标准差矩阵,计算相关系数矩阵.

```
x' * x/(n−1)      (n=length(longley))
ans=
```

$$
\begin{matrix} \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \times & \times & \times & \times & \times & \times & \times \end{matrix} \quad (16\times 7 \text{阶})
$$

其中相关系数 ×≈1.

先不对 longley 标准化，直接求相关系数：

```
corrcoef(longley)↙
ans=
```

$$
\begin{matrix} \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \times & \times & \times & \times & \times & \times & \times \end{matrix} \quad (16\times 7 \text{阶})
$$

其中相关系数“×”最小值为−0.177 4，最大值为1.000 0.

1.3.4 多项式拟合

下面介绍多项式拟合的两种方法.

1)解超定方程组 $Ax=b$(对 x 不作非负的限制)

设实测数据 $(t_i,y_i)_{i=\overline{1,m}}$，令拟合多项式

$$
y(t)=a_0t^n+a_1t^{n-1}+\cdots+a_{n-1}t+a_n.
$$

将 (t_i,y_i) 代入上式，得

$$
y(t_i)=a_0t_i^n+a_1t_i^{n-1}+\cdots+a_{n-1}t_i+a_n\overset{\text{令}}{=}y_i,\quad i=1,2,\cdots,m,
$$

从而形成超定方程组 $Ax=b$，其中

$$
A=\begin{bmatrix} t_1^n & t_1^{n-1} & \cdots & t_1 & 1 \\ t_2^n & t_2^{n-1} & \cdots & t_2 & 1 \\ \vdots & \vdots & & \vdots & \vdots \\ t_m^n & t_m^{n-1} & \cdots & t_m & 1 \end{bmatrix}_{m\times(n+1)},
$$

$$
x=(a_0,a_1,\cdots,a_{n-1},a_n)^{T},
$$

$$
b=(y_0,y_1,\cdots,y_{n-1},y_n)^{T}.
$$

该超定方程组的最小二乘解，即极小问题 $\min\limits_{x\in R^{n+1}}\|Ax-b\|_2^2$ 的解 x^* 为所求的拟合系数

$$
x^*=(a_0^*,a_1^*,\cdots,a_n^*)^{T}.
$$

实现上述过程的 Matlab 程序步骤为：

第一步，形成超定方程组的系数矩阵 A，即

```
for j=1:(n+1)
A(:,j)=t.^(n+1-j);
end
```

第二步，解超定方程组 $Ax=b$，即

```
x=A\b
```

注 超定方程的系数矩阵 A 是超定矩阵，因而不能定义逆阵，但可定义广义逆阵. 这里取 A 的左广义逆阵 $A_L^+=(A^TA)^{-1}A^T$，故 $x=A_L^+b=(A^TA)^{-1}A^Tb$，即

```
x=inv(A'A) * A' * b;
```

下面编写 Matlab 程序：

Matlab 函数 lsqnonlin('f',x)

```
x=lsqnonlin('f',x)
```

设实测数据(t_i,y_i)，$i=0,1,\cdots,10$，如下表：

t_i	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
y_i	2.1	2.3	2.5	2.9	3.2	3.3	3.8	4.1	4.9	5.4	5.8

用三次多项式拟合上述实测数据($n=3$).

下面编写 Matlab 程序：

(1)定义文本文件 fitting.m.

```
t=0 : 0.1 : 1;
b=[2.1  2.3  2.5  2.9  3.2  3.3  3.8  4.1  4.9  5.4  5.8];
n=input('input n:');
for j=1 : (n+1)
A(:,j)=t'.^(n+1-j);
end
x=A\b
```

(2)存盘 fitting.m.

(3)运行文本文件：

```
fitting
```

屏幕显示：

```
input n : 3
```

屏幕显示：

```
x=
   0.6993
   1.2005
   1.8869
   2.1077
```

(4)若需画出拟合函数的图形，只需在文本文件 fitting.m 中加入以下内容：

```
t=0.05 : 0.1 : 1.05;(0.05 为初值,0.1 为步长,1.05 为终值)
u=polyval(x,t);
Plot(t,u,s,b,'+');
```

在上述各语句中，各标志变量的含义为：

x——实测数据自变元值；

u——拟合多项式的值；

b——实测数据函数值；

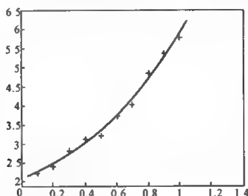
s——拟合系数向量；

t——等距原点.

(5)运行.

fitting

屏幕显示图形如附录图-3,图中实线代表拟合曲线,"+"代表实测数据的散点.



附录图-3

2)直接调用 Matlab 内部函数

直接调用 Matlab 内部函数

polyfit(x,y,n)

其中参数向量 x,y 表示实测数据,n 表示拟合多项式的次数.

(1)建立文本文件 fitting. m.

```
t=0:0.1:1;
b=[2.1  2.3  2.5  2.9  3.2  3.3  3.8  4.1  4.9  5.4  5.8];
n=input('input n:')
x=polyfit(t,b,n)
```

(2)存盘 fitting. m.

(3)运行:

Fitting

屏幕显示:

input n:3

屏幕显示:

```
x=
    0.6993
    1.2005
    1.8869
    2.1077
```

1.3.5 多元线性回归

为处理例 3 中的经济数据，现考虑六元线性回归问题. 设实测数据如下：

$$(t_{i1}, t_{i2}, t_{i3}, t_{i4}, t_{i5}, t_{i6}, y_i), \quad i=\overline{1,m},$$

其中回归自变元 $t_1, t_2, \cdots, t_6$ 以及回归因变元 y 均为随机变量. 设六元线性回归方程

$$\tilde{y}=\hat{a}_1 t_1+\hat{a}_2 t_2+\hat{a}_3 t_3+\hat{a}_4 t_4+\hat{a}_5 t_5+\hat{a}_6 t_6+\hat{a}_7,$$

即对于 $t_1, t_2, \cdots, t_6$ 的每一组值，有

$$y \sim N(a_1 t_1+a_2 t_2+\cdots+a_6 t_6+a_7, \sigma^2),$$

其中 $a_i (i=\overline{1,6})$，σ^2 均为未知参数，且不依赖 $t_i (i=\overline{1,6})$，亦即

$$y=a_1 t_1+\cdots+a_6 t_6+a_7+\varepsilon, \quad \varepsilon \sim N(0, \sigma^2).$$

下面利用所给实测数据去估计参数 $a_i (i=\overline{1,6})$ 的值 $\hat{a}_i (i=\overline{1,6})$，从而得到回归方程

$$\tilde{y}=\hat{a}_1 t_1+\hat{a}_2 t_2+\cdots+\hat{a}_6 t_6+\hat{a}_7.$$

参数 $\hat{a}_i (i=\overline{1,6})$ 的估计，采用极大似然法.

记

$$\begin{aligned} t_1 &= [t_{11} \quad t_{21} \quad \cdots \quad t_{m1}] \\ t_2 &= [t_{12} \quad t_{22} \quad \cdots \quad t_{m2}] \\ &\vdots \\ t_6 &= [t_{16} \quad t_{26} \quad \cdots \quad t_{m6}], \\ y &= [y_1 \quad y_2 \quad \cdots \quad y_m], \\ \bar{x} &= [\hat{a}_1 \quad \hat{a}_2 \quad \cdots \quad \hat{a}_6, \hat{a}_7]. \end{aligned}$$

从而形成超定方程组的系数矩阵

$$A=\begin{bmatrix} t_{11} & t_{12} & \cdots & t_{16} & 1 \\ t_{21} & t_{22} & \cdots & t_{26} & 1 \\ \vdots & \vdots & & \vdots & \vdots \\ t_{m1} & t_{m2} & \cdots & t_{m6} & 1 \end{bmatrix}_{m\times 7}$$

由超定方程组 $Ax=y$ 解得

$$x=A/y'$$

例 5 给出 16×7 的数据块 longley. 现以 longley 的前 6 列数据(年份，GNP，GDP，失业率，军队人数，人口数)线性回归第 7 列数据——就业率.

解 (1)标准化 longley，得单位标准差矩阵.

x=x • /(e * std(x))

(2)令：

y=x(:,7)

(3)形成超定方程组的系数矩阵.

A=[x(:,1:6)　ones(y)]

(4)形成超定方程组.

A $\bar{x}$=y

(5)求解：

```
x̄=A/y'
```

注 $\bar{x}=(\hat{a}_1,\hat{a}_2,\cdots,\hat{a}_6,\hat{a}_7)^T$ 代表回归方程的参数向量的估计值，即为问题之所求。

下面用 Matlab 编程：

(1)建立文本文件 coef.m.

```
e=ones(length(longley),1);
x=longley-e*mean(longley);
x=x./(e*std(x));
y=x(:,7);
A=[x(:,1:6)  ones(y)];
x1=A\y
```

(2)存盘 coef.m.

(3)运行：

```
coef ↙
```

屏幕显示：

```
x1 =
     1.0e+003 *
     0.0018
     0.0000
     0.0000
     0.0000
     0.0000
     0.0000
    -3.4982
```

1.4 数值分析常用算法

Algo.1 一元 n 点 Lagrange 插值

1)方法简介

给定实测数据 $(x_i,y_i)_{i=\overline{1,n}}$，其中 $x_i\neq x_j\,(i\neq j)$，$y_i=f(x_i)$，利用 Lagrange 插值公式计算插值点 x 对应的函数值

$$y(x)=\sum_{k=1}^{n}y_k l_k(x),$$

其中

$$l_k(x)=\prod_{\substack{j=1\\j\neq k}}^{n}\frac{x-x_j}{x_k-x_j}.$$

2)函数语句及参数说明

函数语句为

$$y=\mathrm{lagr}(x_0,y_0,n,x)$$

调用 M 文件 lagr.m，其中各参数说明如下：

x_0—— n 个下标元素的一维实型数组,输入参数,存放给定的插值节点.

y_0—— n 个下标元素的一维实型数组,输入参数,存放插值节点相应的函数值.

n ——整型变量,输入参数,节点个数.

x ——实型变量,输入参数,插值节点.

y ——实型变量,输出参数,插值结果.

3)子程序 lagr. m(在 Matlab 编辑空间写入并存盘)

```
function  y=lagr(x0,y0,n,x)
  y=0,0;
  for  k=1:n
    p=1.0;
     for  j=1:n
       if  j~=k
              p=p*(x-x0(y))/x0(k)-x0(j);
         end
     end
       y=p*y0(k)+y;
     end
```

4)主程序

方式 1 在 Matlab 命令空间发布如下命令:

```
x0=[×  ×  ×];
y0=[×  ×  ×];
n=×;     x=×;
y=lagr(x0,y0,n,×)↙
```

屏幕显示:

```
y=
  ×
```

注 ×代表输入数据或输出数据,一个×代表一个数,三个×代表多个数.

方式 2 在 Matlab 编辑空间写入 M 文件 L. m:

```
x0=[×  ×  ×];
y0=[×  ×  ×];
n=×;  x=×;
y=lagr(x0,y0,n,x)
```

存盘 L. m,然后在 Matlab 命令空间发布命令:

```
L. m↙
```

屏幕显式:

```
y=
  ×
```

例 6 已知 $f(x)-e^{-x^2}$ 的数据如附表 1-2 所示,试用三次插值计算 $x=0.35, 0.55$ 处

的函数值.

附表 1-2

x_0	−0.6	−0.4	−0.2	0	0.2	0.4	0.6
y_0	0.697 676	0.852 114	0.960 789	1	0.960 789	0.852 114	0.697 676

解 采用方式 1,在 Matlab 命令空间发布如下指令:

```
x0=[-0.6  -0.4  -0.2  0.0  0.2  0.4  0.6];
y0=[0.697676  0.852114  0.960789  1.000000  0.960789  0.852114
    0.697676];
n=3;
y=lagr(x0,y0,n,0.35)↙
```

屏幕显示:

```
y=
   0.885305
y=lagr(x0,y0,n,0.55)↙
```

屏幕显示:

```
y=
   0.739279
```

Algo. 2 Romberg 方法求积(逐次半分加速)

1)方法简介

用逐次半分加速方法计算定积分的近似值,其中:

$\{T_0^{(k)}\}_{k=0,1,\cdots}$代表梯形积分值序列,

$\{T_1^{(k)}\}_{k=0,1,\cdots}$代表 Simpson 积分值序列,

$\{T_2^{(k)}\}_{k=0,1,\cdots}$代表 Cotes 积分值序列,

$\{T_3^{(k)}\}_{k=0,1,\cdots}$代表 Romberg 积分值序列,

⋮

梯形积分值序列递推公式以及各个积分值序列之间的加速组合公式见本书 4.3 Romberg 算法设计.

2)函数语句及参数说明

函数语句为

$$[R,n]=\text{rombg}(a,b,eps)$$

调用文件 rombg.m,其中各参数说明如下:

a ——实型输入参数,积分下限.

b ——实型输入参数,积分上限.

eps ——实型输入参数,精度(正小数).

R ——实型输入参数,积分结果.

n——实型输入参数,标志当 $n>0$ 时求积成功;当 $n<0$ 时积分值不满足精度,但步长 h 已为零.

3)在 rombg.m 中所调用的函数 $y=f(x)$ 是由用户自定义的函数,用 f.m 文件表示.

4)子程序 rombg.m

```
function[R.n]=rombg(a,b,eps)
R=10;
R0=0;
k-2;
t(1,1)=b-a*(f(a)+f(b))/2;
while(abs(R-R0)>eps)
  h=(b-a)/2^(k-1);
  w=0;
if(h~=0)
    for i=1:(2^(k-1)-1)
        w=w+f(a+i*h)
    end
t(k,1)=h*(f(a)/2+w+f(b)/2);
for j=2:k
  for i=1:(k-j+1)
    t(i,3)=4^(j-1)*t(i+1,j-1)-t(i,j-1)…/(4^(j-1)-1);
end
end
R=t(1,k);
R0=t(1,k-1);
  k=k+1;
    n=k;
  else   R=R0;
      n=-k;
  end
end
```

5)主程序

```
a=×;b=×;eps=×;
[R,n]=rombg(a,b,eps)↙
```

屏幕显示:

```
R=
   ×
n=
   ×
```

例 7 计算积分 $I=\int_0^1 \frac{4}{1+x^2}\mathrm{d}x$ 的近似值.

解 在 Matlab 命令空间发布如下指令：

```
a=0;b=1;eps=0.5×10^-5;
[R,n]=rombg(a,b,eps)↙
```

屏幕显示：

```
R=
   3.141592653638244e+000
n=
   7
```

注 (1)R 的精确值为 π；$n=7$ 表示精度满足，积分成功；被积函数求值次数为 8.

(2)f.m 文件为

```
function y=f(x)
y=4/(1+x^2)
```

Algo. 3 Gauss－Seidel 迭代法

1)方法简介

设 $Ax=b,\quad A=(a_{ij})_{n\times n},\quad b=(b_1,b_2,\cdots,b_n)^{\mathrm{T}}$.

用 Gauss－Seidel 迭代法解 $Ax=b$ 的算法详见本书 4.2.3.

2)函数语句及参数说明

函数语句为

S=seidel(a,b,x,n,eps)

调用文件 seidel.m，其中参数说明如下：

a —— $n\times n$ 个下标元素的二维实型数组，输入参数；

b —— n 个下标元素的一维实型数组，输入参数；

x —— n 个下标元素的一维实型数组，最初存放初始迭代向量，最后存放解向量；

n ——整型输入参数，线性方程组的阶数；

eps ——实型输入参数，精度(正小数).

3)子程序 seidel.m

```
function S=seidel(a,b,x,n,eps)
k=1;
eorr=0;
for i=1:n
t=x(i);x(i)=0;
x(i)=(b(i)-a(i,:)*n)/a(i,i);
eorr=max(abs(t-x(i)),eorr);
end
  if(abs(eorr)>eps)k=k+1;
```

```
      return;
    end
      s=x;
      k;
    eorr
```

4)主程序

```
    a=[×  ×  ×];
    b=[×  ×  ×];
    x=[×  ×  ×];
    n=×;
    eps=×;
    S=seidel(a,b,x,n,eps)↙
```

屏幕显示：

```
    S=
       ×  ×  ×
    eorr=
         ×
```

例 8 用 Gauss—Seidel 迭代法解 $Ax=b$，其中

$$A=\begin{bmatrix}5 & -1 & -1 & -1\\ -1 & 10 & -1 & -1\\ -1 & -1 & 5 & -1\\ -1 & -1 & -1 & 10\end{bmatrix},\quad b=(-4,12,8,34)^T.$$

解 在 Matlab 命令空间发布如下指令：

```
    a=[5  -1  -1  -1;-1  10  -1  -1;-1  -1  5  -1;-1  -1  -
       1  10];
    b=[-4  12  8  34];
    x=[0  0  0  0];
    n=4;  eps=0.0005;
    S=seidel(a,b,x,n,eps)↙
    x=
       0.99897849430002
       1.99958456867649
       2.99953139743435
       3.99980944604109
    k=
       8
    eorr
         1.697029148829410×10^-4
```

注　迭代 8 次，$\| x^{(8)} - x^{(17)} \|_\infty$ = eorr.

Algo. 4　n 次代数方程求根指令

设实系数 n 次方程

$$x^n + a_1 x^{n-1} + a_2 x^{n-2} + \cdots + a_{n-1} x + a_n = 0,$$

其系数为 $n+1$ 维向量

$$p = [1, a_1, a_2, \cdots, a_{n-1}, a_n].$$

在 Matlab 命令空间直接发布命令

roots(p)

可求得所给 n 次方程的根. 该指令所调用的程序模块是根据多项式友阵的特征根的方法编写的.

例 9　求以下 8 次方程的根.

$$x^8 - 36x^7 + 546x^6 - 4\,536x^5 + 22\,449x^4 - 67\,284x^3 + 118\,124x^2 - 109\,584x + 40\,320 = 0.$$

解　发布以下指令:

```
p=[1  -36  546  -4536  22449  -67284  118124  -109584  40320];
roots(p)↙
```

屏幕显示:

```
ans=
     8  7  6  5  4  3  2  1
```

即方程的 8 个根依次为

$$x_1 = 8,\quad x_2 = 7,\quad x_3 = 6,\quad x_4 = 5,\quad x_5 = 4,\quad x_6 = 3,\quad x_7 = 2,\quad x_8 = 1.$$

若将 7 次项的系数修改为 −37, 发布指令:

```
p(2)=-37↙
```

屏幕显示:

```
p=
  1  -37  546  -4536  22449  -67284  118124  -109584  40320
```

发布指令:

```
roots(p)↙
```

屏幕显示:

```
ans=
     16.1192
     5.0351+5.149i
     5.0351  5.149i
     2.08210+1.7281i
     2.8210-1.7281i
     2.8244+0.2494i
     2.8244-0.2494i
     0.9998
```

第二个方程的8个根中，除了2个实根16.119 2和0.999 8外，其余6个根都是复根. 可见，一个多项式的系数有微小变化对方程的根有很大的影响.

Algo. 5　n 次代数方程求根的 Newton 迭代法

1)方法简介

设实系数 n 次代数方程

$$f(x)=a_0x^n+a_1x^{n-1}+\cdots+a_{n-1}x+a_n=0.$$

用 Newton 迭代法求该方程在初始点 x_0 附近的一个根，算法详见本书 2.3.3.

2)函数语句及参数说明

函数语句为

$$y=newton(a,n,x_0,n_0,eps)$$

调用 M 文件 newton.m，其中参数说明如下：

a —— $n+1$ 个下标元素的一维实型数组，输入参数，按降幂存放方程系数.

n ——输入整型参数，方程的阶数.

x_0——输入实型参数，迭代初值.

n_0——输入整型参数，最大迭代次数.

eps ——输入实型参数，控制根的精度.

3)子程序 newton.m

```
function y=newton(a,n,x0,n0,eps)
  x(1)=x0;
    b=1;
    i=1;
  while(abs(b)>eps * x(i))
    i=i+1
  x(i)=x(i-1)-f(a,n,x(i-1))/df(a,n,x(i-1));
    b=x(i)-x(i-1);
  if(i>n0)error('n0   is   full');
    return;
  end
end
  y=x(i);
    i
```

4)M 文件 newton.m 的调用

在 M 文件 newton.m 中要调用两个函数语句，一个用 f.m 文件存盘，用以计算 n 次多项式的值，即

```
function   y=f(a,n,x)
  y=0.0;
  for i=1 : (n+1)
```

```
        y=y+a(i) * x^(n+1-i);
      end
```

另一个用 df.m 文件存盘,用以计算 n 次多项式的一阶导数值.

```
function y=df(a,n,x)
y=0.0;
for i=1 : n
y=y+a(i) * (n+1-i) * x^(n-i);
end
```

5)主程序

```
a=[×  ×  ×];
n=×;
x0=×;
eps=×;
y=newton(a,n,x0,n0,eps)↙
```

屏幕显示:

```
y=
   ×(方程的根)
i=
   ×(迭代次数)
```

例 10 求 $f(x)=x^3+2x^2+10x-20$ 在 $x_0=1$ 附近的根.

解 发布如下指令:

```
a=[1  2  10  -20];
x0=1;
n0=100;
eps=1e-8;
y=newton(a,n,x0,n0,eps)
```

屏幕显示:

```
y=
   1.368808107821373e+000
i=
   6
```

Algo.6 一阶常微分方程初值问题求解

1)方法简介

设
$$\begin{cases} y'=f(t,y), \\ y(t_0)=y_0, \quad t\in[a,b]. \end{cases}$$

求解上述一阶常微分方程初值问题的 Runge-Kutta 方法详见本书 8.5.1.

2)函数语句及参数说明

函数语句为

$$[t,y]=\text{ode23}(\text{'yprime'},to,tfinal,y_0,tol)$$

调用 M 文件 ode23.m,其中各参数说明如下：

yprime ——输入参数,由用户提供的方程的右函数文件名字串.例如右函数为 fun(t,y),则 yprime=fun.

to —— t 的起始值,输入实型参数.

tfinal —— t 的终值,输入实型参数.

y_0—— 初始值,输入实型参数.

tol —— 输入参数,控制解的精度(缺省值:tol=1e-6).

t ——输出参数,根据精度需求,算法自动选取自变量的结点值(列向量).

y ——输出参数,最终计算所得方程的数值解(列向量).

3)子程序 ode23

注 *ode23 表示用二阶和三阶 Runge-Kutta 公式求解初值问题(程序略);ode45 表示用四阶和五阶 Runge-Kutta 公式求解初值问题,是一个高精度方法,且其使用方法与 ode23 相同.*

例 11 Multhus 及 Logistic 人口模型计算.

在本书 8.1.1 中分别列出了上述两个人口模型

$$\begin{cases} y' = ry, \\ y(x_0) = y, \end{cases} \quad 及 \quad \begin{cases} y' = ry\left(1-\dfrac{y}{k}\right), \\ y(x_0) = y_0. \end{cases}$$

在上述两组公式中,r 为人口自然增长率(r=1.4%);k 为环境容纳量(设 k=36 亿).设 1982 年我国人口为 10.154 1 亿,试用上述两种不同的人口模型计算从 1980 年至 2050 年共 70 年内我国人口的变化数据,并给出对应的增长曲线.

解 首先建立两个方程不同的右函数

```
function Z=peop1(x,y)
Z=0.014*y;
```

及

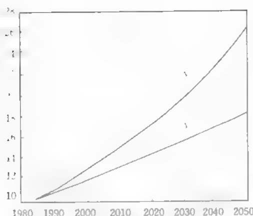
```
function Z=peop2(x,y)
Z=0.014*y*(1-y/36);
```

并在 Matlab 编辑空间分别用两个 M 文件 peop1.m 及 peop2.m 存盘.

主程序 在 Matlab 命令空间发布命令：

```
[t1,y1]=ode23('peop1',1982,2050,10.1541);
[t2,y2]=ode23('peop2',1982,2050,10.1541);
plot(t1,y1,t2,y2)↙
```

屏幕显示 70 年内我国人口变化数据及对应的增长曲线,如附录图 4 所示.



附录图 4

计算结果显示，2[illegible]年中国人口的数量按 Malthus 模型预测值 y[illegible]大约为 26 亿，而按 Logistic 模型预测值 y_2 大约为 18 亿.

Algo. [illegible]阶常微分方程组初值问题求解

1)方法简介

设一阶方程组的初值问题

$$y' = f(t, y),$$

$$y(t[illegible]) [illegible] y[illegible], \quad t [illegible] a, b.$$

其中 $y(t) = (y_1(t), y_2(t), \cdots, y_n(t))^T$, $y(t_0) = (y_{10}, y_{20}, \cdots, y_{n0})^T$.

求解上述方程组初值问题的算法见本书 8.1.1.

2)函数语句及参数说明

函数语句为

```
[t,y]=ode23('yprime',to,tfinal,y0,tol)
```

其中各参数说明与 Algo[illegible]中的函数语句基本一样，只是"y"及"y"的内容有区别.

y_0——输入参数，列向量，存放初始向量.

y[illegible]输[illegible]参数，[illegible] n 组解[illegible]$y_n(t)$(均为列向量)

例[illegible] Lotk[illegible]V[illegible]terr[illegible]捕食者和被捕食者模型. 本书[illegible]中列出了该模型的[illegible]阶常微分方程组

$$\frac{dy}{dx} [illegible] y [illegible] \alpha y [illegible],$$

$$\frac{dy_2}{dx} = -y_2 + \beta y_1 y_2.$$

其中 y[illegible]分别代表被捕食者(野兔)和捕食者(狐狸)的数量

取参数

$$\alpha = 0.01, \quad \beta = 0.02, \quad x \in (0, 15).$$

初值

$$y_1(0)=20, \quad y_2(0)=20.$$

试计算时间范围在(0,15)内,两个种群的数量变化及相应的生长曲线.

解 首先建立上述方程组的右端函数.

```
function yp=lotka(x,y)
yp=diag([1-0.01*y(2),-1+0.02*y(1)]*y);
```

在 Matlab 编辑空间存盘 M 文件.

```
lotka.m
```

主程序 在 Matlab 命令空间发布命令:

```
to=0;
tfinal=15*(1+eps);
y0=[20  20]';
[t,y]=ode23('lotka',to,tfinal,y0);
plot(t,y)↙
```

屏幕显示被捕食者及捕食者两个种群的数量变化规律的图形(见本书 8.6.2 图 8-4).

附录 Ⅱ　习题详解及参考答案

习题 2 解答

1. 用二分法求下列方程的根，要求绝对误差限为 0.005.

(1) $x^3 - x + 4 = 0$，求根区间为 $(-2,-1)$.

解　1° 确定二分次数 K：

因为绝对误差限　$\varepsilon = 0.005 = 0.5 \times 10^{-2}$，

所以　$$K = \left[\frac{\lg \frac{b-a}{2\varepsilon}}{\lg 2}\right] = \left[\frac{\lg \frac{-1+2}{2\times 0.005}}{\lg 2}\right] \approx [6.6] = 7;$$

2° 列表计算：

$f(-2) = (-2)^3 - (-2) + 4 = -2 < 0$

$f(-1) = (-1)^3 - (-1) + 4 = 4 > 0$　　$(-2, -1)$

①$f(-1.5) > 0$　　$(-2, -1)$

②$f(-1.75) > 0$　　$(-2, -1.75)$

③$f(-1.875) < 0$　　$(-1.875, -1.75)$

④$f(-1.812\,58) < 0$　　$(-1.812\,5, -1.75)$

⑤$f(-1.781\,25) > 0$　　$(-1.812\,5, -1.781\,25)$

⑥$f(-1.796\,875) < 0$　　$(1.796\,875, -1.781\,25)$

⑦$f(-1.789\,062\,5) > 0$　　$(-1.796\,875, -1.789\,062\,5)$

令　$$x^* = \frac{-1.796\,875 - 1.789\,062\,5}{2} = -1.792\,968\,75 \approx -1.79.$$

(2) $2x = e^{-x}$，求根区间为 $(0,1)$.

解　令　$f(x) = 2x - e^{-x} = 0$，

绝对误差限　$\varepsilon = 0.005 = 0.5 \times 10^{-2}$.

1° 确定二分次数 K：

$$K = \left[\frac{\lg \frac{b-a}{2\varepsilon}}{\lg 2}\right] = \left[\frac{\lg \frac{1-0}{2\times 0.005}}{\lg 2}\right] \approx [6.6] = 7;$$

2° 列表计算：

$f(0) = -1 < 0$

$f(1) = 1.63 > 0$　　$(0,1)$

①$f(0.5) > 0$　　$(0,0.5)$

②$f(0.25) < 0$　　$(0.25,0.5)$

③$f(0.375) > 0$　　$(0.25,0.375)$

④$f(0.3125)<0$ (0.3125, 0.375)

⑤$f(0.34375)<0$ (0.34375, 0.375)

⑥$f(0.359375)>0$ (0.34375, 0.359375)

⑦$f(0.3515625)<0$ (0.35156255, 0.359375)

令 $x^* = (0.3515625 + 0.359375)/2 \approx 0.35546875 \approx 0.36.$

2. 用二分法搜索方程 $6x^3 - 4x - 1 = 0$ 的实根分布情况，初始搜索区间为$(-2,2)$，并求出(0,1)中的根，精确到0.05.

解 在区间$[-2,2]$上，取步长为1，计算 $f(x) = 6x^3 - 4x - 1$ 的函数值，列表如下：

x	-2	-1	0	1	2
y	41	-3	1	1	39

可见，在区间[0,1]上有一实根. 据题设所求，近似根的绝对误差上限为

$$\varepsilon = 0.05 = 0.5 \times 10^{-1},$$

1° 确定二分次数 K：

$$K = \left[\frac{\lg \frac{b-a}{2\varepsilon}}{\lg 2}\right] = \left[\frac{\lg \frac{1-0}{2\times 0.05}}{\lg 2}\right] \approx [3.3] = 4;$$

2° 列表计算：

$f(0) = -1 < 0$

$f(1) = 1 > 0$ [0,1]

①$f(0.5) < 0$ [0.5,1]

②$f(0.75) < 0$ [0.75,1]

③$f(0.875) < 0$ [0.875,1]

④$f(0.9375) > 0$ [0.875,0.9375]

令 $x^* = (0.875 + 0.9375)/2 = 0.90625 \approx 0.9.$

3. 验证本章例6中迭代公式 $x_{k+1} = \dfrac{x_k^3 + 1}{3}$ 不收敛于1.5和−2附近的根.

解 验证迭代公式不收敛于1.5附近的根. 取 $x_0 = 1.5$，列表计算如下：

$x_0 = 1.5$ $x_6 = 0.394061625$ $x_{12} = 0.347296523$

$x_1 = 1.458333333$ $x_7 = 0.353730562$ $x_{13} = 0.347296375$

$x_2 = 1.367163387$ $x_8 = 0.348086882$ $x_{14} = 0.347296357$

$x_3 = 1.185137978$ $x_9 = 0.347391921$ $x_{15} = 0.347296355$

$x_4 = 0.888195982$ $x_{10} = 0.347307885$ $x_{16} = 0.347296355$

$x_5 = 0.566896932$ $x_{11} = 0.347297746$ $x_{17} = 0.347296355$

验证迭代公式不收敛于−2附近的根. 取 $x_0 = -2$，计算如下：

$x_0 = -2$ $x_1 = -2.333333333$

$x_2 = -3.901234568$ $x_3 = -19.45845039$

$x_4 = -2455.526067$ ……………………

由计算结果可知，该迭代公式不收敛于1.5及−2附近的根.

4. 用迭代法求 $x^3 - 2x - 5 = 0$ 的正根. 下面有三种迭代公式：

(1)$x=\dfrac{x^3-5}{2}$；　(2)$x=\dfrac{5}{x^2-2}$；　(3)$x=\sqrt[3]{2x+5}$.

先判定它们在 $x_0=2$ 附近的收敛情况，并选择收敛的方法求出此根，要求结果精确到 4 位小数.

解　首先判定各迭代公式在 $x_0=2$ 附近的收敛情况.

(1) 令
$$\varphi(x)=\frac{x^3-5}{2},\quad \varphi'(x)=\frac{3x^2}{2}.$$

显然 $\varphi(x)$ 在 $x_0=2$ 附近连续.

因为
$$\varphi'(x_0)=\varphi'(2)=\frac{3\times 2^2}{2}=6>1,$$

故收敛判别法失效，但通过实际计算可知迭代公式 $x=\dfrac{x^3-5}{2}$ 在 $x_0=2$ 附近不收敛.

(2) 令
$$\varphi(x)=\frac{5}{x^2-2},\quad \varphi'(x)=\frac{-2x\times 5}{(x^2-2)^2}=-\frac{10x}{(x^2-2)^2}.$$

显然 $\varphi(x)$ 在 $x_0=2$ 附近连续.

因为
$$\varphi'(x_0)=\varphi'(2)=-\frac{10\times 2}{(2^2-2)^2}=-5,$$
$$|\varphi'(x_0)|>1,$$

故收敛判别法失效，但通过实际计算知迭代公式 $x=\dfrac{5}{x^2-2}$ 在 $x_0=2$ 附近不收敛.

(3) 令
$$\varphi(x)=\sqrt[3]{2x+5}=(2x+5)^{1/3},$$
$$\varphi'(x)=\frac{1}{3}(2x+5)^{-\frac{2}{3}}\times 2=\frac{2}{3}(2x+5)^{-\frac{2}{3}}=\frac{2}{3\sqrt[3]{(2x+5)^2}}.$$

显然 $\varphi(x)$ 在 $x_0=2$ 附近连续.

因为
$$\varphi'(x_0)=\varphi'(2)=\frac{2}{3\sqrt[3]{(2\times 2+5)^2}}=0.154<1,$$

所以迭代公式 $x=\sqrt[3]{2x+5}$ 在 $x_0=2$ 附近收敛.

下面用迭代公式 $x=\sqrt[3]{2x+5}$ 列表计算求方程的根.

因为所求根要求精确到 4 位小数，所以
$$\varepsilon=\frac{1}{2}\times 10^{-4}=0.000\,05.$$

列表计算如下：

k	x_k	$\lvert x_{k+1}-x_k\rvert$
0	2	
1	2.080 083 823	0.080 083 823
2	2.092 350 678	0.012 266 854
3	2.094 216 996	0.001 866 318
4	2.094 500 652	0.000 283 656
5	2.094 543 758	0.000 043 105

显然 $|x_{k+1}-x_k|=0.000\,043\,105<0.000\,05$，故 $x^*=2.094\,543\,758\approx 2.094\,5$.

5. 用松驰法将第 4 题迭代公式(3) 加速，并求 x_0 　2 附近的根，使其精确到 4 位小数.

解　原方程的根要求精确到 4 位小数，所以取绝对误差上限

$$\varepsilon = \frac{1}{2} \times 10^{-4} = 0.00005.$$

因为
$$\varphi'(x) = \frac{2}{3}(2x+5)^{-\frac{2}{3}},$$

故
$$L_k = \varphi'(x_k) = \frac{2}{3}(2x_k+5)^{-\frac{2}{3}}.$$

加速迭代公式(松驰公式) 为

$$\begin{cases} \bar{x}_{k+1} = \varphi(x_k) = \sqrt[3]{2x_k+5}, \\ x_{k+1} = \omega_k \bar{x}_{k+1} + (1-\omega_k)x_k, \\ \omega_k = \dfrac{1}{1-L_k}. \end{cases}$$

列表计算如下：

k	x_k	$\varphi(x_k)$	L_k	$\lvert x_{k+1}-x_k \rvert$
0	2	2. 080 083 23	0. 154 080 283	—
1	2. 094 670 012	2. 094 569 493	0. 151 956 469	0. 094 670 012
2	2. 094 551 482	2. 094 551 482	0. 151 959 082	0. 000 118 53
3	2. 094 551 482	—	—	0. 000 000 000

所以 $x^* = 2.094\,551\,482 \approx 2.094\,6$.

6. 用迭代法计算 $x = \cos x$ 的正根收敛较慢. 试用 Aitken 方法加速收敛，求此根，使其精确到 4 位小数，并与迭代法的收敛速度比较.

解：　因要求 $x = \cos x$ 的正根精确到 4 位小数，所以取绝对误差上限

$$\varepsilon = \frac{1}{2} \times 10^{-4} = 0.000\,05.$$

方法 1　简单迭代法.

取 $x_0 = 10^\circ$ 列表计算如下：

k	x_k	$\lvert x_{k+1}-x_k \rvert$
0	10	
1	0. 984 807 75	
2	0. 999 852 29	0. 015 044 54
3	0. 999 847 74	0. 000 004 55

所以
$$x^* = 0.999\,847\,74 \approx 0.999\,8.$$

方法 2 Aitken 加速迭代.

取 $x_0 = 10^{\circ}$ 列表计算如下：

k	x_k	$\bar{x}_{k+1}$	$\tilde{x}_{k+1}$	$\lvert x_{k+1} - x_k \rvert$
0	10	0.984 807 753	0.999 852 287	—
1	0.999 827 223	0.999 847 747	0.999 847 741	9.000 172 777
2	0.999 847 741	—	—	0.000 020 518

所以 $x^* = 0.999\ 847\ 741 \approx 0.999\ 8$.

7. 用 Newton 迭代法：

(1) 不开平方计算 $\sqrt{10}$，且使其结果精确到 5 位小数.

解 令
$$x = \sqrt{10}, \quad 即\ x^2 - 10 = 0,$$
再令
$$f(x) = x^2 - 10 = 0,$$
故 Newton 迭代公式为
$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{x_k^2 - 10}{2x_k} = \frac{1}{2}\left(x_k + \frac{10}{x_k}\right),$$
即
$$x_{k+1} = \frac{1}{2}\left(x_k + \frac{10}{x_k}\right).$$

取 $x_0 = 3, \varepsilon = \frac{1}{2} \times 10^{-5} = 0.000\ 005$，列表计算如下：

k	x_k	$\lvert x_{k+1} - x_k \rvert$
0	3	
1	3.166 666 667	0.166 666 667
2	3.162 280 702	0.004 385 964
3	3.162 277 66	0.000 003 041

所以 $x^* = 3.162\ 277\ 66 \approx 3.162\ 28$.

(2) 计算方程 $x^3 - x - 1 = 0$ 在 1.3 附近的根，精确到 5 位小数.

解 据题设，近似根的绝对误差上限为
$$\varepsilon = 0.5 \times 10^{-5} = 0.000\ 005.$$
令
$$f(x) = x^3 - x - 1 = 0,$$
Newton 迭代公式为
$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)},$$
其中
$$f(x_k) = x_k^3 - x_k - 1, \quad f'(x_k) = 3x_k^2 - 1.$$

列表计算如下：

k	x_k	$f(x_k)$	$f'(x_k)$	$\lvert x_{k+1} - x_k \rvert$
0	1.3	−0.103	4.07	—
1	1.325 307 125	0.002 513 965	4.269 316 929	0.025 307 125
2	1.324 718 281	0.000 001 378	4.264 635 568	0.000 588 844
3	1.324 717 957	—	—	0.000 000 323

所以 $x^* = 1.324\,717\,957 \approx 1.324\,72.$

(3) 计算 $2x - e^{-x}$ 的根,精确到 4 位小数,并与 1 题中(2) 的收敛速度作比较.

解 据题设,近似根的绝对误差上限为

$$\varepsilon = 0.5 \times 10^{-4} = 0.000\,05.$$

令

$$f(x) = 2x - e^{-x} = 0.$$

Newton 迭代公式为

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)},$$

其中

$$f(x_k) = 2x_k - e^{-x_k}, \qquad f'(x_k) = 2 + e^{-x_k}.$$

列表计算如下:

k	x_k	$\lvert x_{k+1} - x_k \rvert$
0	1	—
1	0.310 724 807	0.689 275 193
2	0.351 511 258	0.040 786 451
3	0.351 733 704	0.000 222 446
4	0.351 733 711	0.000 000 007

所以 $x^* = 0.351\,733\,711 \approx 0.351\,7.$

习题 3 解答

1. 用 Gauss 顺序消元法求解下列方程组 $\boldsymbol{Ax}=\boldsymbol{b}$.

(1) $\boldsymbol{A}=\begin{bmatrix}1 & 1 & -1\\ 1 & 2 & -2\\ -2 & 1 & 1\end{bmatrix}$, $\boldsymbol{b}=\begin{bmatrix}1\\0\\1\end{bmatrix}$.

解

$$(\boldsymbol{A}\mid\boldsymbol{b})=\left[\begin{array}{ccc|c}1 & 1 & -1 & 1\\ 1 & 2 & -2 & 0\\ -2 & 1 & 1 & 1\end{array}\right]\begin{matrix}\\ m_{21}=1\\ m_{31}=-2\end{matrix}\xrightarrow{(1)}\left[\begin{array}{ccc|c}1 & 1 & -1 & 1\\ 0 & 1 & -1 & -1\\ 0 & 3 & -1 & 3\end{array}\right]_{m_{32}=3}$$

$$\xrightarrow{(2)}\left[\begin{array}{ccc|c}1 & 1 & -1 & 1\\ 0 & 1 & -1 & -1\\ 0 & 0 & 2 & 6\end{array}\right],$$

$x_3=6/2=3$,

$x_2=-1+1\times3=2$,

$x_1=1+1\times3-1\times2=2$.

所以 $\boldsymbol{x}^*=[x_1\quad x_2\quad x_3]^{\mathrm{T}}=(2\quad 2\quad 3)^{\mathrm{T}}$.

(2) $\boldsymbol{A}=\begin{bmatrix}4 & 3 & 2 & 1\\ 3 & 4 & 3 & 2\\ 2 & 3 & 4 & 3\\ 1 & 2 & 3 & 4\end{bmatrix}$, $\boldsymbol{b}=\begin{bmatrix}1\\1\\-1\\-1\end{bmatrix}$.

$$(\boldsymbol{A}\mid\boldsymbol{b})=\left[\begin{array}{cccc|c}4 & 3 & 2 & 1 & 1\\ 3 & 4 & 3 & 2 & 1\\ 2 & 3 & 4 & 3 & -1\\ 1 & 2 & 3 & 4 & -1\end{array}\right]\begin{matrix}\\ m_{21}=3/4\\ m_{31}=1/2\\ m_{41}=1/4\end{matrix}$$

$$\xrightarrow{(1)}\left[\begin{array}{cccc|c}4 & 3 & 2 & 1 & 1\\ 0 & 7/4 & 3/2 & 5/4 & 1/4\\ 0 & 3/2 & 3 & 5/2 & -3/2\\ 0 & 5/4 & 5/2 & 15/4 & -5/4\end{array}\right]\begin{matrix}\\ \\ m_{32}=6/7\\ m_{42}=5/7\end{matrix}$$

$$\xrightarrow{(2)}\left[\begin{array}{cccc|c}4 & 3 & 2 & 1 & 1\\ 0 & 7/4 & 3/2 & 5/4 & 1/4\\ 0 & 0 & 12/7 & 10/7 & -12/7\\ 0 & 0 & 10/7 & 20/7 & -10/7\end{array}\right]_{m_{43}=5/6}$$

$$\xrightarrow{(3)}\left[\begin{array}{cccc|c}4 & 3 & 2 & 1 & 1\\ 0 & 7/4 & 3/2 & 5/4 & 1/4\\ 0 & 0 & 12/7 & 10/7 & -12/7\\ 0 & 0 & 0 & 35/21 & 0\end{array}\right],$$

$$x_4 = 0/(35/21) = 0,$$

$$x_3 = (-12/7 - 0 \times 10/7)/(12/7) = -1,$$

$$x_2 = [1/4 - (5/4) \times 0 - 3/2(-1)]/(7/4) = 1,$$

$$x_1 = [1 - 1 \times 0 - 2 \times (-1) - 3 \times 1]/4 = 0.$$

所以 $\boldsymbol{x}^* = (x_1, x_2, x_3, x_4)^{\mathrm{T}} = (0, 1, -1, 0)^{\mathrm{T}}$.

2. 用列主元消元法解下列方程组 $\boldsymbol{Ax} = \boldsymbol{b}$.

(1) $\boldsymbol{A} = \begin{bmatrix} -3 & 2 & 6 \\ 10 & -7 & 0 \\ 5 & -1 & 5 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 4 \\ 7 \\ 6 \end{bmatrix}.$

解

$$(\boldsymbol{A} \vdots \boldsymbol{b}) = \begin{bmatrix} -3 & 2 & 6 & \vdots & 4 \\ 10 & -7 & 0 & \vdots & 7 \\ 5 & -1 & 5 & \vdots & 6 \end{bmatrix} \xrightarrow{r_1 \leftrightarrow r_2} \begin{bmatrix} 10 & -7 & 0 & \vdots & 7 \\ -3 & 2 & 6 & \vdots & 4 \\ 5 & -1 & 5 & \vdots & 6 \end{bmatrix} \begin{matrix} \\ m_{21} = -3/10 \\ m_{31} = 5/10 = 1/2 \end{matrix}$$

$$\xrightarrow{(1)} \begin{bmatrix} 10 & -7 & 0 & \vdots & 7 \\ 0 & -1/10 & 6 & \vdots & 61/10 \\ 0 & 5/2 & 5 & \vdots & 5/2 \end{bmatrix}$$

$$\xrightarrow{r_2 \leftrightarrow r_3} \begin{bmatrix} 10 & -7 & 0 & \vdots & 7 \\ 0 & 5/2 & 5 & \vdots & 5/2 \\ 0 & -1/10 & 6 & \vdots & 61/10 \end{bmatrix} m_{32} = (-1/10)/(5/2) = -1/25$$

$$\xrightarrow{(2)} \begin{bmatrix} 10 & -7 & 0 & \vdots & 7 \\ 0 & 5/2 & 5 & \vdots & 5/2 \\ 0 & 0 & 31/5 & \vdots & 31/5 \end{bmatrix},$$

$$x_3 = (31/5)/(31/5) = 1,$$

$$x_2 = (5/2 - 5 \times 1)/(5/2) = -1,$$

$$x_1 = [7 - 0 \times 1 - (-7) \times (-1)]/10 = 0.$$

所以 $\boldsymbol{x}^* = (x_1, x_2, x_3)^{\mathrm{T}} = (0, -1, 1)^{\mathrm{T}}$.

(2) $\boldsymbol{A} = \begin{bmatrix} 0 & 2 & 0 & 1 \\ 2 & 2 & 3 & 2 \\ 4 & -3 & 0 & 1 \\ 6 & 1 & -6 & -5 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 0 \\ -2 \\ -7 \\ 6 \end{bmatrix}.$

解

$$(\boldsymbol{A} \vdots \boldsymbol{b}) = \begin{bmatrix} 0 & 2 & 0 & 1 & \vdots & 0 \\ 2 & 2 & 3 & 2 & \vdots & -2 \\ 4 & -3 & 0 & 1 & \vdots & -7 \\ 6 & 1 & -6 & -5 & \vdots & 6 \end{bmatrix}$$

$$\xrightarrow{r_1 \leftrightarrow r_4} \begin{bmatrix} 6 & 1 & -6 & -5 & \vdots & 6 \\ 2 & 2 & 3 & 2 & \vdots & -2 \\ 4 & -3 & 0 & 1 & \vdots & -7 \\ 0 & 2 & 0 & 1 & \vdots & 0 \end{bmatrix} \begin{matrix} \\ m_{21} = 2/6 = 1/3 \\ m_{31} = 4/6 = 2/3 \\ m_{41} = 0/6 = 0 \end{matrix}$$

$$\xrightarrow{(1)}\left[\begin{array}{cccc:c}6 & 1 & -6 & -5 & 6\\ 0 & 5/3 & 5 & 11/3 & -4\\ 0 & -11/3 & 4 & 13/3 & -11\\ 0 & 2 & 0 & 1 & 0\end{array}\right]$$

$$\xrightarrow{r_2\leftrightarrow r_3}\left[\begin{array}{cccc:c}6 & 1 & -6 & -5 & 6\\ 0 & -11/3 & 4 & 13/3 & -11\\ 0 & 5/3 & 5 & 11/3 & -4\\ 0 & 2 & 0 & 1 & 0\end{array}\right]\begin{array}{l}m_{32}=(5/3)/(-11/3)=-5/11\\ m_{42}=2/(-11/3)=-6/11\end{array}$$

$$\xrightarrow{(2)}\left[\begin{array}{cccc:c}6 & 1 & -6 & -5 & 6\\ 0 & -11/3 & 4 & 13/3 & -11\\ 0 & 0 & 75/11 & 62/11 & 9\\ 0 & 0 & 24/11 & 37/11 & -6\end{array}\right]m_{43}=(24/11)/(75/11)=8/25$$

$$\xrightarrow{(3)}\left[\begin{array}{cccc:c}6 & 1 & -6 & -5 & 6\\ 0 & -11/3 & 4 & 13/3 & -11\\ 0 & 0 & 74/11 & 62/11 & -9\\ 0 & 0 & 0 & 39/25 & -78/25\end{array}\right],$$

$x_4=(-78/25)/(39/25)=-2,$

$x_3=[-9-62/11\times(-2)]/(75/11)=1/3,$

$x_2=[-11-13/3\times(-2)-4\times 1/3]/(-11/3)=1,$

$x_1=[6+5\times(-2)+6\times 1/3-1\times 1]/6=-1/2.$

所以 $\boldsymbol{x}^*=(x_1,x_2,x_3,x_4)^{\mathrm{T}}=(-1/2,1,1/3,-2)^{\mathrm{T}}$.

3. 用 Gauss－Jordan 消元法求 $\begin{bmatrix}1 & 1 & -1\\ 2 & 1 & 0\\ 1 & -1 & 0\end{bmatrix}^{-1}$.

解法 1

$$\left[\begin{array}{ccc:ccc}1 & 1 & -1 & 1 & 0 & 0\\ 2 & 1 & 0 & 0 & 1 & 0\\ 1 & -1 & 0 & 0 & 0 & 1\end{array}\right]\xrightarrow{(1)}\left[\begin{array}{ccc:ccc}1 & 1 & -1 & 1 & 0 & 1\\ 0 & -1 & 2 & -2 & 1 & 0\\ 0 & -2 & 1 & -1 & 0 & 1\end{array}\right]$$

$$\xrightarrow{(2)}\left[\begin{array}{ccc:ccc}1 & 0 & 1 & -1 & 1 & 0\\ 0 & 1 & -2 & 2 & -1 & 0\\ 0 & 0 & -3 & 3 & -2 & 1\end{array}\right]$$

$$\xrightarrow{(3)}\left[\begin{array}{ccc:ccc}1 & 0 & 0 & 0 & 1/3 & 1/3\\ 0 & 1 & 0 & 0 & 1/3 & -2/3\\ 0 & 0 & 1 & -1 & 2/3 & -1/3\end{array}\right].$$

验证：

$$\begin{bmatrix}1 & 1 & -1\\ 2 & 1 & 0\\ 1 & -1 & 0\end{bmatrix}\begin{bmatrix}0 & 1/3 & 1/3\\ 0 & 1/3 & -2/3\\ 1 & 2/3 & -1/3\end{bmatrix}=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}.$$

所以
$$\begin{bmatrix}1 & 1 & -1\\ 2 & 1 & 0\\ 1 & -1 & 0\end{bmatrix}^{-1} = \begin{bmatrix}0 & 1/3 & 1/3\\ 0 & 1/3 & 2/3\\ -1 & 2/3 & -1/3\end{bmatrix}.$$

解法 2

$$\left[\begin{array}{ccc|ccc}1 & 1 & -1 & 1 & 0 & 0\\ 2 & 1 & 0 & 0 & 1 & 0\\ 0 & -1 & 0 & 0 & 0 & 1\end{array}\right] \xrightarrow{r_1 \leftrightarrow r_2} \left[\begin{array}{ccc|ccc}2 & 1 & 0 & 0 & 1 & 0\\ 1 & 1 & -1 & 1 & 0 & 0\\ 1 & -1 & 0 & 0 & 0 & 1\end{array}\right]$$

$$\xrightarrow{(1)} \left[\begin{array}{ccc|ccc}1 & 1/2 & 0 & 0 & 1/2 & 0\\ 0 & 1/2 & -1 & 1 & -1/2 & 0\\ 0 & -3/2 & 0 & 0 & -1/2 & 1\end{array}\right] \xrightarrow{r_2 \leftrightarrow r_3} \left[\begin{array}{ccc|ccc}1 & 1/2 & 0 & 0 & 1/2 & 0\\ 0 & 3/2 & 0 & 0 & -1/2 & 1\\ 0 & 1/2 & -1 & 1 & -1/2 & 0\end{array}\right]$$

$$\xrightarrow{(2)} \left[\begin{array}{ccc|ccc}1 & 0 & 0 & 0 & 1/3 & 1/3\\ 0 & 1 & 0 & 0 & 1/3 & -2/3\\ 0 & 0 & -1 & 1 & -2/3 & 1/3\end{array}\right] \xrightarrow{(3)} \left[\begin{array}{ccc|ccc}1 & 0 & 0 & 0 & 1/3 & 1/3\\ 0 & 1 & 0 & 0 & 1/3 & -2/3\\ 0 & 0 & 1 & -1 & 2/3 & -1/3\end{array}\right].$$

所以

$$\begin{bmatrix}1 & 1 & -1\\ 2 & 1 & 0\\ 1 & -1 & 0\end{bmatrix}^{-1} = \begin{bmatrix}0 & 1/3 & 1/3\\ 0 & 1/3 & -2/3\\ -1 & 2/3 & -1/3\end{bmatrix}.$$

4. 用直接 **LU** 分解法求解第 1 题中的两个矩阵的 **LU** 分解，并求解这两个方程组.

(1) $\boldsymbol{A} = \begin{bmatrix}1 & 1 & -1\\ 1 & 2 & -2\\ -2 & 1 & 1\end{bmatrix}$, $\boldsymbol{b} = \begin{bmatrix}1\\ 0\\ 1\end{bmatrix}$.

解 列表计算如下：

$u_{11} = 1$　　$u_{12} = 1$　　$u_{13} = 1$

$L_{21} = 1/1 = 1$　　$u_{22} = 2 - 1 \times 1 = 1$　　$u_{23} = -2 + 1 \times 1 = -1$

$L_{31} = -2/1 = -2$　　$L_{32} = [1 - (-2 \times 1)]/1 = 3$　　$u_{33} = 1 - (-2) \times (-1) - 3 \times (-1) = 2$

所以

$$\boldsymbol{L} = \begin{bmatrix}1 & 0 & 0\\ 1 & 1 & 0\\ 2 & 3 & 1\end{bmatrix}, \quad \boldsymbol{U} = \begin{bmatrix}1 & 1 & -1\\ 0 & 1 & -1\\ 0 & 0 & 2\end{bmatrix}.$$

将 $\boldsymbol{A} = \boldsymbol{LU}$ 代入原方程组得

$$\boldsymbol{LUx} = \boldsymbol{b}.$$

令 $\boldsymbol{Ux} = \boldsymbol{y}$ 得三角方程组

$$\begin{cases}\boldsymbol{Ly} = \boldsymbol{b},\\ \boldsymbol{Ux} = \boldsymbol{y}.\end{cases}$$

其中
$$\boldsymbol{Ly} = \boldsymbol{b},$$

即
$$\begin{bmatrix}1 & 0 & 0\\ 1 & 1 & 0\\ -2 & 3 & 1\end{bmatrix}\begin{bmatrix}y_1\\ y_2\\ y_3\end{bmatrix} = \begin{bmatrix}1\\ 0\\ 1\end{bmatrix}.$$

解此三角方程组得

$y_1 = 1,$

$y_2 = 0 - y_1 = -1,$

$y_3 = 1 + 2 \times y_1 - 3 \times y_2 = 1 + 2 \times 1 + 3 \times 1 = 6.$

因 $\boldsymbol{Ux} = \boldsymbol{b}$，即

$$\begin{bmatrix} 1 & 1 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 6 \end{bmatrix}.$$

解此三角方程组得

$x_3 = 6/2 = 3,$

$x_2 = -1 + x_3 = -1 + 3 = 2,$

$x_1 = 1 - x_2 + x_3 = 1 - 2 + 3 = 2.$

所以 $\boldsymbol{x}^* = (x_1 \quad x_2 \quad x_3)^{\mathrm{T}} = (2 \quad 2 \quad 3)^{\mathrm{T}}.$

$$(2)\boldsymbol{A} = \begin{bmatrix} 4 & 3 & 2 & 1 \\ 3 & 4 & 3 & 2 \\ 2 & 3 & 4 & 3 \\ 1 & 2 & 3 & 4 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}.$$

解 列表计算如下.

$U_{1行}: u_{11} = 4 \qquad u_{12} = 3 \qquad u_{13} = 2 \qquad u_{14} = 1$

$L_{列1}: L_{21} = 3/4 \qquad L_{31} = 2/4 = 1/2 \qquad L_{41} = 1/4$

$U_{2行}: u_{22} = 4 - 9/4 = 7/4 \qquad u_{23} = 3 - 6/4 = 3/2 \qquad u_{24} = 2 - 3/4 = 5/4$

$L_{列2}: L_{32} = (3 - (1/2) \times 3)(7/4) = 6/7 \qquad L_{42} = (2 - (1/4) \times 3)/(7/4) = 5/7$

$U_{3行}: u_{33} = 4 - (1/2) \times 2 - (6/7) \times (3/2) = 12/7$

$u_{34} = 3 - (1/2) \times 1 - (6/7) \times (5/4) = 10/7$

$L_{列3}: L_{43} = [3 - (1/4) \times 2 - (5/7) \times (3/2)]/(12/7) = 5/6$

$U_{4行}: u_{44} = 4 - (1/4) \times 1 - (5/7) \times (5/4) - (5/6) \times (10/7) = 5/3$

$$所以 \quad \boldsymbol{L} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 3/4 & 1 & 0 & 0 \\ 1/2 & 6/7 & 1 & 0 \\ 1/4 & 5/7 & 5/6 & 1 \end{bmatrix}, \quad \boldsymbol{U} = \begin{bmatrix} 4 & 3 & 2 & 1 \\ 0 & 7/4 & 3/2 & 5/4 \\ 0 & 0 & 12/7 & 10/7 \\ 0 & 0 & 0 & 5/3 \end{bmatrix}.$$

将 $\boldsymbol{A} = \boldsymbol{LU}$ 代入原方程组得

$$\boldsymbol{LUx} = \boldsymbol{b}.$$

令 $\boldsymbol{Ux} = \boldsymbol{y}$，得三角方程组

$$\begin{cases} \boldsymbol{Ly} = \boldsymbol{b}, \\ \boldsymbol{Ux} = \boldsymbol{y}. \end{cases}$$

因 $\boldsymbol{Ly} = \boldsymbol{b}$，即

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 3/4 & 1 & 0 & 0 \\ 1/2 & 6/7 & 1 & 0 \\ 1/4 & 5/7 & 5/6 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \end{bmatrix},$$

解得

$$y_1 = 1,$$
$$y_2 = 1 - 3/4 = 1/4,$$
$$y_3 = -1 - (1/2)y_1 - (6/7)y_2$$
$$= 1 - 1/2 - (6/7) \times (1/4) = -12/7,$$
$$y_4 = -1 - (1/4)y_1 - (5/7)y_2 - (5/6)y_3$$
$$= -1 - 1/4 - (5/7) \times (1/4) - (5/6) \times (-12/7) = 0.$$

因 $\boldsymbol{Ux} = \boldsymbol{y}$，即

$$\begin{bmatrix} 4 & 3 & 2 & 1 \\ 0 & 7/4 & 3/2 & 5/4 \\ 0 & 0 & 12/7 & 10/7 \\ 0 & 0 & 0 & 5/3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1/4 \\ -12/7 \\ 0 \end{bmatrix},$$

解得

$$x_4 = 0,$$
$$x_3 = (-12/7 - 0 \times 10/7)/(12/7) = -1,$$
$$x_2 = [1/4 - 0 \times 5/4 - (-1) \times 3/2] \times (7/4) = 1,$$
$$x_1 = [1 - 1 \times 0 - 2 \times (-1) - 3 \times 1]/4 = 0.$$

所以 $\boldsymbol{x}^* = (x_1, x_2, x_3, x_4)^T = (0, 1, -1, 0)^T$.

5. 用追赶法解三对角方程组 $\boldsymbol{Ax} = \boldsymbol{b}$，其中：

$$\boldsymbol{A} = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

解 设 $\boldsymbol{b} = \boldsymbol{f} = (1 \quad 0 \quad 0 \quad 0 \quad 0)^T$，列表计算如下：

1	1	2	3	4	5
c_i	−1	−1	−1	−1	—
b_i	2	2	2	2	2
a_i	—	−1	−1	−1	−1
f_i	1	0	0	0	0
α_i	2	3/2	4/3	5/4	6/5
β_i	−1/2	−2/3	3/4	−4/5	—

[illegible]

y_i	1/2	1/3	1/4	1/5	1/6
x_i	5/6	2/3	1/2	1/3	1/6

故 $x=(\frac{5}{6}\quad\frac{2}{3}\quad\frac{1}{2}\quad\frac{1}{3}\quad\frac{1}{6})^T$.

习题 4 解答

1. 用 Jacobi 迭代法和 Seidel 迭代法求解下列方程组(要求解向量的各分量准确至三位小数).

(1) $\begin{bmatrix}2 & 1 & 0\\1 & 2 & 1\\0 & 1 & 2\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}=\begin{bmatrix}3\\-5\\4\end{bmatrix}$; (2) $\begin{bmatrix}-8 & 1 & 1\\1 & -5 & 1\\1 & 1 & -4\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}=\begin{bmatrix}1\\16\\7\end{bmatrix}$.

用 Jacobi 迭代法求解.

解 (1)① 构造 Jacobi 迭代矩阵:

$$B_J=-D^{-1}(L+U),$$

即 $$B_J=\begin{bmatrix}-0.5 & 0 & 0\\0 & -0.5 & 0\\0 & 0 & -0.5\end{bmatrix}\begin{bmatrix}0 & 1 & 0\\1 & 0 & 1\\0 & 1 & 0\end{bmatrix}=\begin{bmatrix}0 & -0.5 & 0\\-0.5 & 0 & -0.5\\0 & -0.5 & 0\end{bmatrix}.$$

又因为 $$d_J=D^{-1}b,$$

则 $$d_J=\begin{bmatrix}0.5 & 0 & 0\\0 & 0.5 & 0\\0 & 0 & 0.5\end{bmatrix}\begin{bmatrix}3\\-5\\4\end{bmatrix}=\begin{bmatrix}1.5\\-2.5\\2\end{bmatrix}.$$

② 列表计算:

$$\varepsilon=\frac{1}{2}\times 10^{-3}=0.0005.$$

$x^{(k)}$	$B_J=-D^{-1}(L+U)$			$d_J=D^{-1}b$	$\|x^{(k+1)}-x^{(k)}\|_\infty$
	0	−0.5	0	1.5	
	−0.5	0	−0.5	−2.5	
	0	−0.5	0	2	
$x^{(0)}$	1.5	−2.5	2		
$x^{(1)}$	2.75	−4.25	3.25		1.75
$x^{(2)}$	3.625	−5.5	4.125		1.25
$x^{(3)}$	4.25	−6.375	4.75		0.875
$x^{(4)}$	4.6875	−7	5.1875		0.625
$x^{(5)}$	5	−7.4375	5.5		0.4375
$x^{(6)}$	5.21875	−7.75	5.71875		0.3125
$x^{(7)}$	5.375	−7.96875	5.875		0.21875
$x^{(8)}$	5.484375	−8.125	5.984375		0.15625
$x^{(9)}$	5.5625	−8.234375	6.0625		0.109375

续表

$x^{(k)}$	$B_J=-D^{-1}(L+U)$			$d_J=D^{-1}b$	$\Vert x^{(k+1)}-x^{(k)}\Vert_\infty$
$x^{(10)}$	5.617 187 5	−8.312 5	6.117 187 5		0.078 125
$x^{(11)}$	5.656 25	−8.367 187 5	6.156 25		0.054 687 5
$x^{(12)}$	5.683 593 75	−8.406 26	6.183 593 75		0.039 062 5
$x^{(13)}$	5.703 125	−8.433 593 75	6.203 125		0.027 343 75
$x^{(14)}$	5.716 796 875	−8.453 125	6.216 796 875		0.019 531 25
$x^{(15)}$	5.726 562 5	−8.466 796 875	6.226 562 5		0.013 671 875
$x^{(16)}$	5.733 40	−8.476 56	6.234 40		0.009 763 125
$x^{(17)}$	5.738 28	−8.483 4	6.238 28		0.006 84
$x^{(18)}$	5.741 7	−8.488 28	6.241 7		0.004 88
$x^{(19)}$	5.744 14	−8.491 7	6.244 14		0.003 42
$x^{(20)}$	5.745 85	−8.494 14	6.245 85		0.002 44
$x^{(21)}$	5.747 07	−8.495 85	6.247 07		0.001 71
$x^{(22)}$	5.747 925	−8.497 07	6.247 925		0.001 22
$x^{(23)}$	5.748 535	−8.497 925	6.248 535		0.000 855
$x^{(24)}$	5.748 962 5	−8.498 535	6.248 962 5		0.000 61
$x^{(25)}$	5.749 267 5	−8.498 962 5	6.249 267 5		0.000 427 5

因为 $\Vert x^{(25)}-x^{(24)}\Vert_\infty=0.4275\times10^{-3}<0.5\times10^{-3}$，

故 $x^*=x^{(25)}=(5.749,\ -8.499,\ 6.249)^T$.

(2)① 构造 Jacobi 迭代矩阵：

$$B_J=D^{-1}(L+U),$$

即
$$B_J=\begin{bmatrix}1/8&0&0\\0&1/5&0\\0&0&1/4\end{bmatrix}\begin{bmatrix}0&1&1\\1&0&1\\1&1&0\end{bmatrix}=\begin{bmatrix}0&0.125&0.125\\0.2&0&0.2\\0.25&0.25&0\end{bmatrix}.$$

又因为
$$d_J=D^{-1}b,$$

则
$$d_J=\begin{bmatrix}-0.125&0&0\\0&-0.2&0\\0&0&-0.25\end{bmatrix}\begin{bmatrix}1\\16\\7\end{bmatrix}=\begin{bmatrix}-0.125\\-3.2\\-1.75\end{bmatrix}.$$

② 列表计算：

$$\varepsilon=\frac{1}{2}\times10^{-3}=0.0005.$$

$x^{(k)}$	$B_J=-D^{-1}(L+U)$			$d_J=D^{-1}b$	$\Vert x^{(k+1)}-x^{(k)}\Vert_\infty$
	0	0.125	0.125	−0.125	
	0.2	0	0.2	−3.2	
	0.25	0.25	0	−1.75	

续表

$x^{(k)}$	$B_J=-D^{-1}(L+U)$			$d_J=D^{-1}b$	$\|x^{(k+1)}-x^{(k)}\|_\infty$
$x^{(0)}$	−0.125	−3.2	−1.75		
$x^{(1)}$	−0.743 75	−3.575	−2.581 25		0.831 25
$x^{(2)}$	−0.894 531 25	−3.865	−2.829 687 5		0.29
$x^{(3)}$	−0.961 835 937	−3.944 843 75	−2.939 882 813		0.110 195 313
$x^{(4)}$	−0.984 854 003	−3.980 343 75	−2.976 669 922		0.036 787 109
$x^{(5)}$	−0.994 626 709	−3.992 304 785	−2.991 299 438		0.014 629 516
$x^{(6)}$	−0.997 950 527	−3.997 185 229	−2.996 732 874		0.005 433 436
$x^{(7)}$	−0.999 239 762	−3.998 936 68	−2.998 783 939		0.002 051 065
$x^{(8)}$	−0.999 715 077	−3.999 604 74	−2.999 544 111		0.000 760 171
$x^{(9)}$	−0.999 893 606	−3.999 851 838	−2.999 829 954		0.000 285 843

因为 $\|x^{(9)}-x^{(8)}\|_\infty=0.285\,843\times10^{-3}<0.5\times10^{-3}$,

故 $x^*=x^{(9)}=(-1.000,-4.000,-3.000)^T$

用 Seidel 迭代法求解.

解(1)① 构造 Jacobi 迭代矩阵:

$$B_J=-D^{-1}(L+U),$$

$$B_J=\begin{bmatrix}-0.5&0&0\\0&-0.5&0\\0&0&-0.5\end{bmatrix}\begin{bmatrix}0&1&0\\1&0&1\\0&1&0\end{bmatrix}=\begin{bmatrix}0&-0.5&0\\-0.5&0&-0.5\\0&-0.5&0\end{bmatrix}.$$

又因为 $$d_J=D^{-1}b,$$

则 $$d_J=\begin{bmatrix}0.5&0&0\\0&0.5&0\\0&0&0.5\end{bmatrix}\begin{bmatrix}3\\-5\\4\end{bmatrix}=\begin{bmatrix}1.5\\-2.5\\2\end{bmatrix}.$$

② 列表计算:

$$\varepsilon=\frac{1}{2}\times10^{-3}=0.000\,5.$$

$x^{(k)}$	$B_J=-D^{-1}(L+U)$			$d_J=D^{-1}b$	$\|x^{(k+1)}-x^{(k)}\|_\infty$
	0	−0.5	0	1.5	
	−0.5	0	−0.5	−2.5	
	0	−0.5	0	2	
$x^{(0)}$	1.5	−2.5	2		
$x^{(1)}$	2.75	−4.875	4.437 5		2.437 5
$x^{(2)}$	3.937 5	−6.687 5	5.343 75		1.812 5
$x^{(3)}$	4.843 75	−7.593 75	5.796 875		0.906 25
$x^{(4)}$	5.296 875	−8.046 875	6.023 437 5		0.453 125

续表

$x^{(k)}$	$B_J=-D^{-1}(L+U)$			$d_J=D^{-1}b$	$\lVert x^{(k+1)}-x^{(k)}\rVert_\infty$
$x^{(5)}$	5.523 437 5	−8.273 437 5	6.136 718 5		0.226 562 5
$x^{(6)}$	5.636 718 75	−8.386 718 75	6.193 359 375		0.113 281 25
$x^{(7)}$	5.693 359 375	−8.443 359 375	6.221 679 688		0.056 640 625
$x^{(8)}$	5.721 679 688	8.471 679 688	6.235 839 844		0.028 320 312
$x^{(9)}$	5.735 839 844	−8.485 839 844	6.242 919 922		0.014 160 155
$x^{(10)}$	5.742 919 22	−8.492 919 922	6.246 459 961		0.007 080 078
$x^{(11)}$	5.746 459 961	8.496 459 961	6.248 229 981		0.003 540 039
$x^{(12)}$	5.748 229 981	−8.498 229 981	6.249 114 99		0.001 770 019
$x^{(13)}$	5.749 114 99	−8.499 114 99	6.249 557 495		0.000 885 009
$x^{(14)}$	5.749 557 495	−8.499 557 495	6.249 778 748		0.000 442 504

因为　　$\lVert x^{(14)}-x^{(13)}\rVert_\infty = 0.442\,504\times10^{-3} < 0.5\times10^{-3}$，

故 $x^* = x^{(14)} = (5.749\ -8.500\ -6.250)^T$.

(2)① 构造 Jacobi 迭代矩阵：

$$B=-D^{-1}(L+U),$$

$$B_J=\begin{bmatrix}1/8 & 0 & 0\\ 0 & 1/5 & 0\\ 0 & 0 & 1/4\end{bmatrix}\begin{bmatrix}0 & 1 & 1\\ 1 & 0 & 1\\ 1 & 1 & 0\end{bmatrix}=\begin{bmatrix}0 & 0.125 & 0.125\\ 0.2 & 0 & 0.2\\ 0.25 & 0.25 & 0\end{bmatrix}.$$

又因为　　$d_J=D^{-1}b$,

故　　$$d_J=\begin{bmatrix}-0.125 & 0 & 0\\ 0 & -0.2 & 0\\ 0 & 0 & -0.25\end{bmatrix}\cdot\begin{bmatrix}1\\16\\7\end{bmatrix}=\begin{bmatrix}-0.125\\-3.2\\-1.75\end{bmatrix},$$

② 列表计算：

$$\varepsilon=\frac{1}{2}\times10^{-3}=0.000\,5.$$

$x^{(k)}$	$B_J=-D^{-1}(L+U)$			$d_J=D^{-1}b$	$\lVert x^{(k+1)}-x^{(k)}\rVert_\infty$
	0	0.125	0.125	−0.125	
	0.2	0	0.2	−3.2	
	0.25	0.25	0	−1.75	
$x^{(0)}$	−0.125	−3.2	−1.75		
$x^{(1)}$	−0.743 75	−3.698 75	−2.860 625		1.110 625
$x^{(2)}$	0.944 921 875	3.961 109 375	−2.976 507 813		0.262 359 375
$x^{(3)}$	−0.992 202 148	−3.993 741 992	2.996 486 035		0.047 280 273
$x^{(4)}$	0.998 778 503	3.999 052 908	2.999 457 853		0.006 576 355
$x^{(5)}$	−0.999 813 845	−3.999 854 34	−2.999 917 046		0.001 035 342
$x^{(6)}$	0.999 971 423	−3.999 977 694	−2.999 987 279		0.000 157 578

因为 $\|x^{(6)}-x^{(5)}\|_\infty = 0.157\,578\times 10^{-3} < 0.5\times 10^{-3}$，

故 $x^* = x^{(6)} = (-1.000\quad 4.000\quad -3.000)^T$.

2. 以下列矩阵为系数矩阵构成线性方程组，试判断它们对 Jacobi 迭代法、Seidel 迭代法的收敛性。

(1) $\begin{bmatrix} 5 & 2 & 1 \\ 1 & 3 & 2 \\ 1 & 1 & 2 \end{bmatrix}$.

解 显然该系数矩阵弱超且不可约，故对任意初始点 Jacobi 迭代法、Seidel 迭代法均收敛.

(2) $\begin{bmatrix} 1 & 2 \\ 3 & 2 \end{bmatrix}$.

解 显然该系数矩阵不满足强、弱超条件，故需分别讨论两种迭代法的收敛性.

① 讨论 Jacobi 迭代法：

因为 $$B_J = -D^{-1}(L+U),$$

即 $$B = \begin{bmatrix} -1 & 0 \\ 0 & -1/2 \end{bmatrix}\begin{bmatrix} 0 & 2 \\ 3 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -2 \\ -3/2 & 0 \end{bmatrix},$$

所以 $$\|B_J\|_\infty = 2 > 1.$$

又因为 $$\det(\lambda I - B_J) = \begin{vmatrix} \lambda & 2 \\ 3/2 & \lambda \end{vmatrix} = \lambda^2 - 3 = 0,$$

所以 $$\lambda_{1,2} = \pm\sqrt{3},$$

即 $$\rho(B_J) = \sqrt{3} > 1.$$

故 Jacobi 迭代法不收敛.

② 讨论 Seidel 迭代法：

因为 $$B_S = -(D+L)^{-1}U,$$

即 $$B_S = \begin{bmatrix} -1 & 0 \\ 1.5 & -0.5 \end{bmatrix}\begin{bmatrix} 0 & 2 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -2 \\ 0 & 3 \end{bmatrix},$$

所以 $$\|B_S\|_\infty = 3 > 1.$$

又因为 $$\det(\lambda I - B_S) = \begin{vmatrix} \lambda & 2 \\ 0 & \lambda - 3 \end{vmatrix} = \lambda(\lambda - 3) = 0,$$

所以 $$\lambda_1 = 0,\quad \lambda_2 = 3,$$

即 $$\rho(B_S) = 3 > 1.$$

故 Seidel 迭代法不收敛.

鉴于 $B_S = -(D+L)^{-1}\bar{U}$ 中计算 $(D+L)^{-1}$ 比较困难，可以通过求解方程 $\det[\lambda(D+L)+\bar{U}] = 0$ 来计算 $\rho(B_S)$.

因为 $$\det[\lambda(D+L)+\bar{U}] = \begin{vmatrix} \lambda & 2 \\ 3\lambda & 2\lambda \end{vmatrix} = 2\lambda(\lambda - 3) = 3$$

所以 $$\lambda_1 = 0,\quad \lambda_2 = 3,$$

即 $$\rho(\boldsymbol{B}_S) = 3 > 1.$$

3. 设方程组

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \quad a_{11} \neq 0, \quad a_{22} \neq 0.$$

试证明用 *Jacobi* 迭代法收敛的充要条件是

$$r = \left|\frac{a_{12}a_{21}}{a_{11}a_{22}}\right| < 1.$$

证 因为

$$\boldsymbol{B}_J = -\boldsymbol{D}^{-1}(\boldsymbol{L}+\boldsymbol{U}),$$

则 $$\boldsymbol{B}_J = -\begin{bmatrix} 1/a_{11} & 0 \\ 0 & 1/a_{22} \end{bmatrix}\begin{bmatrix} 0 & a_{12} \\ a_{21} & 0 \end{bmatrix} = \begin{bmatrix} 0 & -a_{12}/a_{11} \\ -a_{21}/a_{22} & 0 \end{bmatrix}.$$

又因为 $$\det(\lambda\boldsymbol{I} - \boldsymbol{B}_J) = \begin{vmatrix} \lambda & a_{12}/a_{11} \\ a_{21}/a_{22} & \lambda \end{vmatrix} = \lambda^2 - \frac{a_{12}a_{21}}{a_{11}a_{22}},$$

所以 $$\rho(\boldsymbol{B}_J) = |\lambda| = \left|\sqrt{\frac{a_{12}a_{21}}{a_{11}a_{22}}}\right|.$$

因为 Jacobi 迭代法收敛的充要条件是

$$\rho(\boldsymbol{B}_J) = \left|\sqrt{\frac{a_{12}a_{21}}{a_{11}a_{22}}}\right| < 1,$$

由此证得 $$r = \left|\frac{a_{12}a_{21}}{a_{11}a_{22}}\right| < 1.$$

4. 设 $\boldsymbol{A} = \begin{bmatrix} 1 & a & a \\ a & 1 & a \\ a & a & 1 \end{bmatrix}$，$a$ 为实数，求：

(1) 若 $\boldsymbol{A}$ 正定，a 的取值范围；

(2) 若简单迭代法收敛，a 的取值范围.

解 (1) 若 $\boldsymbol{A}$ 正定，则有

$$\begin{vmatrix} 1 & a \\ a & 1 \end{vmatrix} > 0, \quad \begin{vmatrix} 1 & a & a \\ a & 1 & a \\ a & a & 1 \end{vmatrix} > 0,$$

所以 $$\begin{cases} 1 - a^2 > 0, \\ 1 + 2a^3 - 3a^2 > 0. \end{cases}$$

即有 $$\begin{cases} (1-a)(1+a) > 0, \\ (1-a)^2(2a+1) > 0. \end{cases}$$

故 $-\frac{1}{2} < a < 1$，即 a 的取值范围为 $(-\frac{1}{2}, 1)$.

(2) 简单迭代法收敛的充要条件是

$$\rho(\boldsymbol{B}_J) < 1.$$

而 $$\boldsymbol{B}_J = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}\begin{bmatrix} 0 & a & a \\ a & 0 & a \\ a & a & 0 \end{bmatrix} = \begin{bmatrix} 0 & -a & a \\ -a & 0 & -a \\ a & -a & 0 \end{bmatrix},$$

$$\det(\lambda I - B_J) = \begin{vmatrix} \lambda & a & a \\ a & \lambda & a \\ a & a & \lambda \end{vmatrix} = (\lambda - a)(\lambda - a)(\lambda + 2a) = 0,$$

解得 $$\lambda_{1,2} = a, \quad \lambda_3 = -2a.$$

所以 $$\rho(B_J) = |\lambda_3| = |2a| < 1.$$

故 $-1/2 < a < 1/2$,即 a 的取值范围为 $(-1/2, 1/2)$.

5. 求下列矩阵的 $\|A\|_1, \|A\|_2, \|A\|_\infty, \rho(A)$.

(1)$A = \begin{bmatrix} 0.6 & 0.5 \\ 0.1 & 0.3 \end{bmatrix}$.

解① 求 $\|A\|_1$ 与 $\|A\|_\infty$:

$$\|A\|_1 = 0.8, \quad \|A\|_\infty = 1.1.$$

② 求 $\|A\|_2$:

因为

$$A^T A = \begin{bmatrix} 0.6 & 0.1 \\ 0.5 & 0.3 \end{bmatrix}\begin{bmatrix} 0.6 & 0.5 \\ 0.1 & 0.3 \end{bmatrix} = \begin{bmatrix} 0.37 & 0.33 \\ 0.33 & 0.34 \end{bmatrix},$$

所以

$$\det(\lambda I - A^T A) = \begin{vmatrix} \lambda - 0.37 & -0.33 \\ -0.33 & \lambda - 0.34 \end{vmatrix} = \lambda^2 - 0.71\lambda + 0.0169 = 0,$$

解得 $$\lambda_1 = 0.685, \quad \lambda_2 = 0.025.$$

故 $$\|A\|_2 = \sqrt{\lambda_1} = \sqrt{0.685} \approx 0.827.$$

③ 求谱半径 $\rho(A)$:

因为 $$(\det(\lambda I - A) = \begin{vmatrix} \lambda - 0.6 & -0.5 \\ -0.1 & \lambda - 0.3 \end{vmatrix} = \lambda^2 - 0.9\lambda + 0.13 = 0,$$

所以 $$\lambda_1 = 0.719, \quad \lambda_2 = 0.181.$$

故 $$\rho(A) = 0.719.$$

(2)$A = \begin{bmatrix} 5 & 7 & 3 \\ 7 & 11 & 2 \\ 3 & 2 & 6 \end{bmatrix}$.

解 ① 求 $\|A\|_1$ 与 $\|A\|_\infty$:

$$\|A\|_1 = 20, \quad \|A\|_\infty = 20.$$

② 求谱半径:

$$\det(\lambda I - A) = \begin{vmatrix} \lambda - 5 & -7 & -3 \\ -7 & \lambda - 11 & -2 \\ -3 & -2 & \lambda - 6 \end{vmatrix} = \lambda^3 - 22\lambda^2 + 89\lambda - 1 = 0.$$

解此方程得最大的根为 $\lambda = 16.66$.

故 $$\rho(A) = 16.66.$$

③ 求二范数 $\|A\|_2$.

因为 A 为实对称矩阵,据第 4 章定理 4.2

$$\rho(A) = \|A\|_2,$$

所以 $$\|A\|_2 = 16.66.$$

习题 5 解答

1. 用规范化幂法计算下列矩阵的主特征值及对应的特征向量，当特征值为 3 位小数稳定时终止迭代.

(1)$A_1 = \begin{bmatrix} 7 & 3 & -2 \\ 3 & 4 & -1 \\ -2 & -1 & 3 \end{bmatrix}$，(2)$A_2 = \begin{bmatrix} 3 & -4 & 3 \\ 4 & 6 & 3 \\ 3 & 3 & 1 \end{bmatrix}$.

解 (1) 取 $V_0 = U_0 = (1,1,1)^T$，则

$$\lambda_1 \approx 9.6058, \quad x_1 \approx (1, 0.6056, -0.3945)^T.$$

(2) 取 $V_0 = U_0 = (1,1,1)^T$，则

$$\lambda_1 \approx 8.86951, \quad x_1 \approx (-0.60422, 1, 0.15094)^T.$$

2. 用原点平移法计算 A 的主特征值，其中

$$A = \begin{bmatrix} -3 & 1 & 0 \\ 1 & -3 & -3 \\ 0 & -3 & 4 \end{bmatrix},$$

取 $p = -4$，当特征值有 3 位小数稳定时终止迭代.

解 取 $V_0 = U_0 = (1,1,1)^T$，则当 $p = -4$ 时经 6 次迭代得

$$\lambda_1 \approx 9.1247.$$

3. 设 $A = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 4 \end{bmatrix}$，用反幂法求 A 的最接近于 $q = 1.2679$ 的特征向量，迭代次数为 2 次.

解 用列主元素消元法实现 $A - qI$ 的三角分解，即

$$I_{1,3}I_{2,3}(A - qI) = LU \qquad (q = 1.2679),$$

其中

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0.7321 & -0.26807 & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 1 & 1.7321 & 1 \\ 0 & 1 & 2.7321 \\ 0 & 0 & 2.9405 \times 10^{-4} \end{bmatrix}.$$

第一次迭代：

由 $UV_1 = U_0 = (1,1,1)^T$ 得

$$V_1 = (12692, -9290.3, 3400.8)^T, \quad U_1 = (1, -1.73198, 0.26795)^T.$$

第二次迭代：

由 $LUV_2 = I_{1,3}, I_{2,3}, U_1$ 得

$$V_2 = (20404, \quad 14937, 54674)^T, \quad U_2 = (1, \quad 0.73206, 0.26796)^T.$$

所以 $$x_2 = U_2 = (1, -0.73206, 0.26796)^T.$$

注 x_2(真解) $= (1, 1-\sqrt{3}, 2-\sqrt{3})^T = (1, -0.73205, 0.26795)^T$.

习题 6 解答

1. 已知函数 $f(x)=e^{-x^2}$ 的数表如下：

x_i	−0.6	−0.4	−0.2	0	0.2	0.4	0.6
y_i	0.697 676	0.852 114	0.960 789	1	0.960 789	0.852 114	0.697 676

试用二次、三次插值，计算 $x=0.35$ 及 $x=0.55$ 时函数的近似值.

解 (1) 使用二次插值.

①$x=0.35$.

方法 1 利用三点(0,1),(0.2,0.960 789),(0.4,0.852 114) 做二次插值.

$$L_2(x)=y_3\frac{(x-x_4)(x-x_5)}{(x_3-x_4)(x_3-x_5)}+y_4\frac{(x-x_3)(x-x_5)}{(x_4-x_3)(x_4-x_5)}+y_5\frac{(x-x_3)(x-x_4)}{(x_5-x_3)(x_5-x_4)}$$

$$=1\times[(0.35-0.2)(0.35-0.4)]/[(0-0.2)(0-0.4)]+0.960\,789\times[(0.35-0)(0.35-0.4)]/[(0.2-0)(0.2-0.4)]+0.852\,114\times[(0.35-0)(0.35-0.2)]/[(0.4-0)(0.4-0.2)]$$

$$=0.885\,795.$$

方法 2 利用三点(0.2,0.960 789),(0.4,0.852 114),(0.6,0.697 676) 做二次插值.

$$L_2(x)=0.960\,789\times[(0.35-0.4)(0.35-0.6)]/[(0.2-0.4)(0.2-0.6)]+0.852\,114\times[(0.35-0.2)(0.35-0.6)]/[(0.4-0.2)(0.4-0.6)]+0.697\,676\times[(0.35-0.2)(0.35-0.4)]/[(0.6-0.2)(0.6-0.4)]$$

$$=0.883\,573\,031.$$

②$x=0.55$.

$$L_2(x)=0.960\,789\times[(0.55-0.4)(0.55-0.6)]/[(0.2-0.4)(0.2-0.6)]+0.852\,114\times[(0.55-0.2)(0.55-0.6)]/[(0.4-0.2)(0.4-0.6)]+0.697\,676\times[(0.55-0.2)(0.55-0.4)]/[(0.6-0.2)(0.6-0.4)]$$

$$=0.740\,575\,781.$$

(2) 使用三次插值.

①$x=0.35$.

方法 1 利用四点(0,1),(0.2,0.960 789),(0.4,0.852 114),(0.6,0.697 676) 做三次插值.

$$L_3(x)=1\times[(0.35-0.2)(0.35-0.4)(0.35-0.6)]/[(0-0.2)(0-0.4)(0-0.6)]+0.960\,789\times[(0.35-0)(0.35-0.4)(0.35-0.6)]/[(0.2-0)(0.2-0.4)(0.2-0.6)]+0.852\,114\times[(0.35-0)(0.35-0.2)(0.35-0.6)]/[(0.4-0)(0.4-0.2)(0.4-0.6)]+0.697\,676\times[(0.35-0)(0.35-0.2)(0.35-0.4)]/[(0.6-0)(0.6-0.2)(0.6-0.4)]$$

$$=0.884\,498\,851.$$

方法 2 利用四点(−0.2,0.960 789),(0,1),(0.2,0.960 789),(0.4,0.852 114)做三次插值.

$$
\begin{aligned}
L_3(x) = {} & 0.960\,789\times[(0.35-0)(0.35-0.2)(0.35-0.4)]/[(-0.2-0)(-0.2-0.2)(-0.2-0.4)] + 1\times[(0.35+0.2)(0.35-0.2)(0.35-0.4)]/[(0+0.2)(0-0.2)(0-0.4)] \\
& + 0.960\,789\times[(0.35+0.2)(0.35-0)(0.35-0.4)]/[(0.2+0.2)(0.2-0)(0.2-0.4)] + 0.852\,114\times[(0.35+0.2)(0.35-0)(0.35-0.2)]/[(0.4+0.2)(0.4-0)(0.4-0.2)] \\
= {} & 0.885\,305\,109.
\end{aligned}
$$

②$x = 0.55$.

$$
\begin{aligned}
L_3(x) = {} & 1\times[(0.55-0.2)(0.55-0.4)(0.55-0.6)]/[(0-0.2)(0-0.4)(0-0.6)] + 0.960\,789\times[(0.55-0)(0.55-0.4)(0.55-0.6)]/[(0.2-0)(0.2-0.4)(0.2-0.6)] \\
& + 0.852\,114\times[(0.55-0)(0.55-0.2)(0.55-0.6)]/[(0.4-0)(0.4-0.2)(0.4-0.6)] + 0.697\,676\times[(0.55-0)(0.55-0.2)(0.55-0.4)]/[(0.6-0)(0.6-0.2)(0.6-0.4)] \\
= {} & 0.739\,279\,632.
\end{aligned}
$$

2. 利用函数 $\sin x$ 在 $x=0,\frac{\pi}{6},\frac{\pi}{4},\frac{\pi}{3}$ 及 $\frac{\pi}{2}$ 处的值,求 $\sin(\frac{\pi}{5})$,并估计误差.

解 首先列出如下数表

x_i	0	$\frac{\pi}{6}$	$\frac{\pi}{4}$	$\frac{\pi}{3}$	$\frac{\pi}{2}$
y_i	0	0.5	0.707 106 781	0.866 025 403	1

$$
\begin{aligned}
\sin\left(\frac{\pi}{5}\right) \approx {} & L_4\left(\frac{\pi}{5}\right) \\
= {} & 0 + 0.5\times\frac{(\frac{\pi}{5}-0)(\frac{\pi}{5}-\frac{\pi}{4})(\frac{\pi}{5}-\frac{\pi}{3})(\frac{\pi}{5}-\frac{\pi}{2})}{(\frac{\pi}{6}-0)(\frac{\pi}{6}-\frac{\pi}{4})(\frac{\pi}{6}-\frac{\pi}{3})(\frac{\pi}{6}-\frac{\pi}{2})} \\
& + 0.707\,106\,78\times\frac{(\frac{\pi}{5}-0)(\frac{\pi}{5}-\frac{\pi}{6})(\frac{\pi}{5}-\frac{\pi}{3})(\frac{\pi}{5}-\frac{\pi}{2})}{(\frac{\pi}{4}-0)(\frac{\pi}{4}-\frac{\pi}{6})(\frac{\pi}{4}-\frac{\pi}{3})(\frac{\pi}{4}-\frac{\pi}{2})} \\
& + 0.866\,025\,403\times\frac{(\frac{\pi}{5}-0)(\frac{\pi}{5}-\frac{\pi}{6})(\frac{\pi}{5}-\frac{\pi}{4})(\frac{\pi}{5}-\frac{\pi}{2})}{(\frac{\pi}{3}-0)(\frac{\pi}{3}-\frac{\pi}{6})(\frac{\pi}{3}-\frac{\pi}{4})(\frac{\pi}{3}-\frac{\pi}{2})} \\
& + 1\times\frac{(\frac{\pi}{5}-0)(\frac{\pi}{5}-\frac{\pi}{6})(\frac{\pi}{5}-\frac{\pi}{4})(\frac{\pi}{5}-\frac{\pi}{3})}{(\frac{\pi}{2}-0)(\frac{\pi}{2}-\frac{\pi}{6})(\frac{\pi}{2}-\frac{\pi}{4})(\frac{\pi}{2}-\frac{\pi}{3})}
\end{aligned}
$$

$$= 0.587\,809\,514$$

估计上述近似值的绝对误差

$$\begin{aligned}
\left|R_4\left(\frac{\pi}{5}\right)\right| &= \left|\frac{f^{(5)}(\xi)}{5!}\omega_5\left(\frac{\pi}{5}\right)\right| \leqslant \left|\frac{\omega_4\left(\frac{\pi}{5}\right)}{5!}\right| \\
&= \left|\frac{\left(\frac{\pi}{5}-0\right)\left(\frac{\pi}{5}-\frac{\pi}{6}\right)\left(\frac{\pi}{5}-\frac{\pi}{4}\right)\left(\frac{\pi}{5}-\frac{\pi}{3}\right)\left(\frac{\pi}{5}-\frac{\pi}{2}\right)}{5\times4\times3\times2\times1}\right| \\
&= 0.000\,034\,002 \\
&= 0.340\,0\times10^{-4}.
\end{aligned}$$

所以近似值 $L_4\left(\frac{\pi}{5}\right) = 0.587\,809\,514$，精确至第四位小数为 0.587 8.

3. 利用以下数表

x_i	0	0.2	0.4	0.6	0.8	1.0
$S(x_i)$	0	0.199 56	0.396 46	0.588 13	0.772 10	0.946 08

计算积分

$$S(x) = \int_0^x \frac{\sin t}{t}\mathrm{d}t$$

当 $S(x) = 0.45$ 时，x 的值.

解　利用反插值函数 $x = L_5(S) = \sum_{i=0}^{5} x_i l_i(s)$，并令 $S = 0.45$，求 $x = L_5(0.45)$ 的值.

$$\begin{aligned}
x &= L_5(0.45) \\
&= 0 + 0.2\times[(0.45-0)(0.45-0.396\,46)(0.45-0.588\,13)(0.45-0.772\,10)(0.45 \\
&\quad -0.946\,08)]/[(0.199\,56-0)(0.199\,56-0.396\,46)(0.199\,56-0.588\,13)(0.199\,56 \\
&\quad -0.772\,10)(0.199\,56-0.946\,08)] + 0.4\times[(0.45-0)(0.45-0.199\,56)(0.45 \\
&\quad -0.588\,13)(0.45-0.772\,10)(0.45-0.946\,08)]/[(0.396\,46-0)(0.396\,46 \\
&\quad -0.199\,56)(0.396\,46-0.588\,13)(0.396\,46-0.772\,10)(0.396\,46-0.946\,08)] \\
&\quad +0.6\times[(0.45-0)(0.45-0.199\,56)(0.45-0.396\,46)(0.45-0.772\,10)(0.45 \\
&\quad -0.946\,08)]/[(0.588\,13-0)(0.588\,13-0.199\,56)(0.588\,13-0.396\,46)(0.588\,13 \\
&\quad -0.772\,10)(0.588\,13-0.946\,08)] + 0.8\times[(0.45-0)(0.45-0.199\,56)(0.45 \\
&\quad -0.396\,46)(0.45-0.588\,13)(0.45-0.946\,08)]/[(0.772\,10-0)(0.772\,10 \\
&\quad -0.199\,56)(0.772\,10-0.396\,46)(0.772\,10-0.588\,13)(0.772\,10-0.946\,08)) \\
&\quad +1.0\times[(0.45-0)(0.45-0.199\,45)(0.45-0.396\,46)(0.45-0.588\,13)(0.45 \\
&\quad -0.772\,10)]/[(0.946\,08-0)(0.946\,08-0.199\,56)(0.946\,08-0.396\,46)(0.946\,08 \\
&\quad -0.588\,13)(0.946\,08-0.772\,10)] \\
&= 0.455\,210\,535.
\end{aligned}$$

4. 试用 Newton 插值求经过点 $(-3,-1)$，$(0,2)$，$(3,-2)$，$(6,10)$ 的三次插值公式.

解　列出均差表：

-3	-1			
0	2	1		
3	-2	$-\frac{4}{3}$	$-\frac{7}{18}$	
6	10	4	$\frac{8}{9}$	$\frac{23}{162}$

所以：

$$N_3(x)=-1+1\times(x+3)+(-\frac{7}{18})\times(x+3)(x-0)+\frac{23}{162}\times(x+3)(x-0)(x-3).$$

故 $N_3(x)=\frac{23}{162}x^3-\frac{7}{18}x^2-\frac{13}{9}x+2.$

5. 给出函数 $f(x)$ 的数表如下：

x_i	0.1	0.2	0.3	0.4	0.5
$f(x_i)$	0.700 10	0.401 60	0.108 10	−0.174 40	−0.437 50

试利用该数表求方程 $f(x)=0$ 在 0.3 与 0.4 之间的根.

解 列出均差表

0.1	0.700 10				
0.2	0.401 60	−2.985			
0.3	0.108 10	−2.935	0.25		
0.4	−0.174 40	−2.825	0.55	1	
0.5	−0.437 50	−2.631	0.97	1.4	1

所以

$$\begin{aligned}f(x)=N_4(x)&=0.700\,10-2.985(x-0.1)+0.25(x-0.1)(x-0.2)+(x\\&\quad-0.1)(x-0.2)(x-0.3)+(x-0.1)(x-0.2)(x-0.3)(x-0.4)\\&=x^4-3x+1.\end{aligned}$$

令 $f(x)=N_4(x)=0$ 得

$$x^4-3x+1=0.$$

取 $x_0=0.35$，用 Newton 迭代法求 $f(x)=0$ 在 0.3 与 0.4 之间的根，并将计算结果列表如下：

$$x_{k+1}=x_k-\frac{x_k^4-3x_k+1}{4x_k^3-3}.$$

k	x_k	$\lvert x_{k+1}-x_k\rvert$
0	0.35	
1	0.337 628 159	0.012 371 841
2	0.337 666 765	0.000 038 606
3	0.337 666 765	0.000 000 000

故 $x^*=0.337\,666\,765.$

6. 试构造一个三次 Hermite 插值多项式，使其满足

$$f(0)=1,\quad f'(0)=0.5,\quad f(1)=2,\quad f'(1)=0.5.$$

解　由题意列表如下：

i	x_i	$f(x_i)$	$f'(x_i)$
0	0	1	0.5
1	1	2	0.5

$$\alpha_0(x) = (1+2\times\frac{x-x_0}{x_1-x_0})(\frac{x-x_1}{x_0-x_1})^2 = (1+2\times\frac{x-0}{1-0})(\frac{x-1}{0-1})^2 = (1+2x)(x-1)^2,$$

同理

$$\alpha_1(x) = (1+2\times\frac{x-x_1}{x_0-x_1})(\frac{x-x_0}{x_1-x_0})^2 = (1+2\times\frac{x-1}{0-1})(\frac{x-0}{1-0})^2 = x^2(3-2x),$$

$$\beta_0(x) = (x-x_0)(\frac{x-x_1}{x_0-x_1})^2 = (x-0)(\frac{x-1}{0-1})^2 = x(x-1)^2,$$

$$\beta_1(x) = (x-x_1)(\frac{x-x_0}{x_1-x_0})^2 = (x-1)(\frac{x-0}{1-0})^2 = (x-1)x^2.$$

因为　$H_3(x) = y_0\alpha_0(x) + y_1\alpha_1(x) + y_0'\beta_0(x) + y_1'\beta_1(x),$

所以　$H_3(x) = -x^3 + 1.5x^2 + 0.5x + 1.$

7. 确定一个不高于四次的多项式 $\varphi(x)$，使得

$$\varphi(0) = \varphi'(0) = 0,\qquad \varphi(1) = \varphi'(1) = \varphi(2) = 1.$$

解　先由 $\varphi(0),\varphi(1),\varphi(2)$ 构造一个 Newton 插值多项式 $N_2(x)$.

列均差表

0	0		
1	1	1	
2	1	0	$-1/2$

因为　$N_2(x) = 0 + 1\times(x-0) + (-1/2)\times(x-0)(x-1) = -\frac{1}{2}x^2 + \frac{3}{2}x,$

令　$p_4(x) = N_2(x) + (ax+b)(x-0)(x-1)(x-2),$

则　$p_4(x) = -\frac{1}{2}x^2 + \frac{3}{2}x + (ax+b)(x-0)(x-1)(x-2).$

$$p_4'(x) = -x + \frac{3}{2} + ax(x-1)(x-2) + (ax-b)(x-1)(x-2)$$
$$+ x(ax+b)(x-2) + x(ax+b)(x-1),$$

$$\begin{cases} p_4'(0) = \frac{3}{2} + 2b = 0, \\ p_4'(1) = \frac{1}{2} - a - b = 1, \end{cases}$$

所以　$a = \frac{1}{4},\quad b = -\frac{3}{4}.$

故　$p_4(x) = \frac{1}{4}x^4 - \frac{3}{2}x^3 + \frac{9}{4}x^2.$

8. 已知下表中 x_i，y_i 的数据及边界条件 $S'(0.25) = 1.0000$，$S'(0.53) = 0.6868$. 试求三次样条插值函数 $S(x)$.

x_i	0.25	0.30	0.39	0.45	0.53
y_i	0.500 0	0.547 7	0.624 5	0.670 8	0.728 0

解 由题意,将有关数据列表如下:

j	x_j	y_j	h_j	μ_j	λ_j	d_j
0	0.25	0.500 0	0.05	—	—	—
1	0.30	0.547 7	0.09	0.642 9	0.357 1	2.754 2
2	0.39	0.624 5	0.06	0.4	0.6	2.413 0
3	0.45	0.670 8	0.08	0.571 4	0.428 6	2.242 1
4	0.53	0.728 0	—	—	—	—

因为
$$d_1 - \mu_1 y_0' - \mu_1 S'(0.25) = 2.754\,2 - 0.642\,9 \times 1.000\,0 = 2.111\,3,$$
$$d_3 - \lambda_3 y_4' = d_3 - \lambda_3 S'(0.53) = 2.242\,4 - 0.428\,6 \times 0.686\,8 = 1.947\,7,$$

所以
$$\begin{bmatrix} 2 & 0.357\,1 & 0 \\ 0.4 & 2 & 0.6 \\ 0 & 0.571\,4 & 2 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix} = \begin{bmatrix} 2.111\,3 \\ 2.413\,0 \\ 1.947\,7 \end{bmatrix},$$
$$m_1 = 0.912\,7, \quad m_2 = 0.800\,4, \quad m_3 = 0.745\,2.$$

即 $\boldsymbol{m} = (m_0, m_1, m_2, m_3, m_4)^{\mathrm{T}} = (1.000, 0.912\,7, 0.800\,4, 0.745\,2, 0.686\,8)^{\mathrm{T}}$.

故

$$S(x) = \begin{cases} S_1(x) = 1.88x^3 - 2.424x^2 + 1.859\,5x + 0.157\,25, & x \in [0.25, 0.30], \\ S_2(x) = 0.793\,8x^3 - 1.445\,5x^2 + 1.565\,7x + 0.186\,7, & x \in [0.30, 0.39], \\ S_3(x) = 0.633\,3x^3 - 1.258\,0x^2 + 1.492\,6x + 0.196\,1, & x \in [0.39, 0.45], \\ S_4(x) = 0.312\,5x^3 - 0.824\,4x^2 + 1.297\,3x + 0.225\,5, & x \in [0.45, 0.53]. \end{cases}$$

9. 使电流通过 2Ω 的电阻,用伏特表测量电阻两端的电压,得到如下数据:

I_i	1	2	4	6	8	10
V_i	1.8	3.7	8.2	12.0	15.8	20.2

试用最小二乘法建立 I 与 V 之间的线性经验公式(该公式对于校正测量所用的伏特表有用).

解 设 $V = V_0 + RI$,将表中数据代人,建立超定方程组
$$V_i = V_0 + RI_i \quad (i = \overline{1,6}).$$

该方程组的矩阵形式为 $\boldsymbol{Ax} = \boldsymbol{y}$,其中
$$\boldsymbol{A} = \begin{bmatrix} 1 & I_1 \\ 1 & I_2 \\ 1 & I_3 \\ 1 & I_4 \\ 1 & I_5 \\ 1 & I_6 \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} V_0 \\ R \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{bmatrix}.$$

列表计算法方程组 $A^TAx = A^Ty$ 中的各元素：

i	I_i	V_i	I_i^2	I_iV_i
1	1	1.8	1	1.8
2	2	3.7	4	7.4
3	4	8.2	16	32.8
4	6	12.0	36	72.0
5	8	15.8	64	126.4
6	10	20.2	100	202
$\sum$	31	61.7	221	442.4

由上表 $$A^TA = \begin{bmatrix} 6 & \sum I_i \\ \sum I_i & \sum I_i^2 \end{bmatrix}, \quad A^Ty = \begin{bmatrix} \sum V_i \\ \sum I_iV_i \end{bmatrix},$$

即 $$\begin{bmatrix} 6 & 31 \\ 31 & 221 \end{bmatrix}\begin{bmatrix} V_0 \\ R \end{bmatrix} = \begin{bmatrix} 61.7 \\ 442.4 \end{bmatrix},$$

解之得 $V_0 = -0.215\,6,\quad R = 2.032\,1.$

故得经验公式 $V = -0.215\,6 + 2.032\,1I.$

10. 求下列超定方程组的最小二乘解

$$\begin{cases} 2x + 4y = 11, \\ 3x - 5y = 3, \\ x + 2y = 6, \\ 2x + y = 7. \end{cases}$$

解 上述超定方程组的矩阵形式为

$$AX = \overline{Y}$$

其中 $$A = \begin{bmatrix} 2 & 4 \\ 3 & -5 \\ 1 & 2 \\ 2 & 1 \end{bmatrix}, \quad X = (x,y)^T, \quad \overline{Y} = (11,3,6,7)^T.$$

该超定方程组的法方程组为

$$A^TAX = A^T\overline{Y}.$$

其中 $$A^TA = \begin{bmatrix} 18 & -3 \\ -3 & 46 \end{bmatrix}, \quad X = (x,y)^T, \quad A^T\overline{Y} = (51,48)^T.$$

解法方程组为 $$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 3.040\,3 \\ 1.241\,8 \end{bmatrix}.$$

11. 已知以下数据：

x_i	-1.00	-0.75	-0.50	-0.25	0	0.25	0.50	0.75	1.00
y_i	0.220 9	0.329 5	0.882 6	1.439 2	2.000 3	2.564 5	3.133 4	3.706 1	4.283 6

试用一次、二次、三次多项式按最小二乘原理拟合以上数据，写出法方程组，并求出最小二

乘拟合的二次多项式.

解 (1) 设一次式为 $y = a + bx$,将表中数据代入,得到超定方程组

$$\begin{cases} 0.2209 = a - b, \\ 0.3295 = a - 0.75b, \\ 0.8826 = a - 0.5b, \\ 1.4392 = a - 0.25b, \\ 2.0003 = a, \\ 2.5645 = a + 0.25b, \\ 3.1334 = a + 0.5b, \\ 3.7061 = a + 0.75b, \\ 4.2836 = a + b. \end{cases}$$

其矩阵形式为

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}.$$

其中

$$\boldsymbol{A} = \begin{bmatrix} 1 & -1 \\ 1 & -0.75 \\ 1 & -0.5 \\ 1 & -0.25 \\ 1 & 0 \\ 1 & 0.25 \\ 1 & 0.5 \\ 1 & 0.75 \\ 1 & 1 \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} a \\ b \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} 0.2209 \\ 0.3295 \\ 0.8826 \\ 1.4392 \\ 2.0003 \\ 2.5645 \\ 3.1334 \\ 3.7061 \\ 4.2836 \end{bmatrix}.$$

列表建立法方程组 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y}$,其中

x_i	y_i	$x_i y_i$	x_i^2
−1.00	0.2209	−0.2209	1.000
−0.75	0.3295	−2471	0.5625
⋮	⋮	⋮	⋮
0.75	3.7061	2.7796	0.5625
1.00	4.2836	4.2836	1.0000
$\sum x_i = 0$	$\sum y_i = 185601$	$\sum x_i y_i = 8.001875$	$\sum x_i^2 = 3.75$

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} = \begin{bmatrix} 9 & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix} \begin{bmatrix} 9 & 0 \\ 0 & 3.75 \end{bmatrix}, \quad \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \end{bmatrix} - \begin{bmatrix} 18.5601 \\ 8.001875 \end{bmatrix}.$$

即

$$\begin{bmatrix} 9 & 0 \\ 0 & 3.75 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 18.5601 \\ 8.001875 \end{bmatrix},$$

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 2.0622 \\ 2.1338 \end{bmatrix}.$$

故得最小二乘拟合多项式为

$$y = 2.0622 + 2.1338x.$$

(2) 设二次多项式为 $y = a_2 + b_2 x + c_2 x^2$，将数据代入，得到超定方程组

$$\begin{cases} 0.2209 = a_2 - b_2 + c_2, \\ 0.3295 = a_2 - 0.75b_2 + 0.5625c_2, \\ 0.8826 = a_2 - 0.5b_2 + 0.25c_2, \\ 1.4392 = a_2 - 0.25b_2 + 0.0625c_2, \\ 2.0003 = a_2, \\ 2.5645 = a_2 + 0.25b_2 + 0.0625c_2, \\ 3.1334 = a_2 + 0.5b_2 + 0.25c_2, \\ 3.7061 = a_2 + 0.75b_2 + 0.5625c_2, \\ 4.2836 = a_2 + b_2 + c_2. \end{cases}$$

其矩阵形式为 $\boldsymbol{AX} = \boldsymbol{Y}$，其中

$$\boldsymbol{A} = \begin{bmatrix} 1 & -1 & 1 \\ 1 & -0.75 & 0.5625 \\ 1 & -0.5 & 0.25 \\ 1 & -0.25 & 0.0625 \\ 1 & 0 & 0 \\ 1 & 0.25 & 0.0625 \\ 1 & 0.5 & 0.25 \\ 1 & 0.75 & 0.5625 \\ 1 & 1 & 1 \end{bmatrix}, \quad \boldsymbol{X} = \begin{bmatrix} a_2 \\ b_2 \\ c_2 \end{bmatrix}, \quad \boldsymbol{Y} = \begin{bmatrix} 0.2209 \\ 0.3295 \\ 0.8826 \\ 1.4392 \\ 2.0003 \\ 2.5645 \\ 3.1334 \\ 3.7061 \\ 4.2836 \end{bmatrix}.$$

该超定方程组的法方程组为 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{AX} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{Y}$.

其中 $$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} = \begin{bmatrix} 9 & 0 & 3.75 \\ 0 & 3.75 & 0 \\ 3.75 & 0 & 2.765625 \end{bmatrix}, \quad \boldsymbol{A}^{\mathrm{T}}\boldsymbol{Y} = \begin{bmatrix} 18.5601 \\ 8.001875 \\ 8.02875625 \end{bmatrix},$$

即 $$\begin{bmatrix} 9 & 0 & 3.75 \\ 0 & 3.75 & 0 \\ 3.75 & 0 & 2.765625 \end{bmatrix} \begin{bmatrix} a_2 \\ b_2 \\ c_2 \end{bmatrix} = \begin{bmatrix} 18.5601 \\ 8.001875 \\ 8.02875625 \end{bmatrix}.$$

解之得 $$\begin{bmatrix} a_2 \\ b_2 \\ c_2 \end{bmatrix} = \begin{bmatrix} 1.9599 \\ 2.1338 \\ 0.2455 \end{bmatrix}.$$

故得最小二乘拟合二次多项式

$$y = 1.9599 + 2.1338x + 0.2455x^2.$$

(3) 设三次多项式为 $y = a_3 + b_3 x + c_3 x^2 + d_3 x^3$，代入数据，得到超定方程组

$$
\begin{cases}
0.2209 = a_3 - b_3 - c_3 - d_3, \\
0.3295 = a_3 - 0.75b_3 + 0.5625c_3 - 0.421875d_3, \\
0.8826 = a_3 - 0.5b_3 + 0.25c_3 - 0.125d_3, \\
1.4392 = a_3 - 0.25b_3 + 0.0625c_3 - 0.015625d_3, \\
2.0003 = a_3, \\
2.5645 = a_3 + 0.25b_3 + 0.0625c_3 + 0.015625d_3, \\
3.1334 = a_3 + 0.5b_3 + 0.25c_3 + 0.125d_3, \\
3.1334 = a_3 + 0.5b_3 + 0.25c_3 + 0.125d_3, \\
3.7061 = a_3 + 0.75b_3 + 0.5625c_3 + 0.421875d_3, \\
4.2836 = a_3 + b_3 + c_3 + d_3.
\end{cases}
$$

其矩阵形式为 $\boldsymbol{AX} = \boldsymbol{Y}$,其中

$$
\boldsymbol{A} = \begin{bmatrix}
1 & -1 & 1 & -1 \\
1 & -0.75 & 0.5625 & -0.421875 \\
1 & -0.5 & 0.25 & -0.125 \\
1 & -0.25 & 0.0625 & -0.015625 \\
1 & 0 & 0 & 0 \\
1 & 0.25 & 0.0625 & 0.015625 \\
1 & 0.5 & 0.25 & 0.125 \\
1 & 0.75 & 0.5625 & 0.421875 \\
1 & 1 & 1 & 1
\end{bmatrix}, \quad
\boldsymbol{X} = \begin{bmatrix} a_3 \\ b_3 \\ c_3 \\ d_3 \end{bmatrix}, \quad
\boldsymbol{Y} = \begin{bmatrix} 0.2209 \\ 0.3295 \\ 0.8826 \\ 1.4392 \\ 2.0003 \\ 2.5645 \\ 3.1334 \\ 3.7061 \\ 4.2836 \end{bmatrix}.
$$

建立法方程组 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{AX} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{Y}$,其中

$$
\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} = \begin{bmatrix}
9 & 0 & 3.75 & 0 \\
0 & 3.75 & 0 & 2.765625 \\
3.75 & 0 & 2.765625 & 0 \\
0 & 2.765625 & 0 & 2.387695313
\end{bmatrix},
$$

$$
\boldsymbol{A}^{\mathrm{T}}\boldsymbol{Y} = \begin{bmatrix} 18.5601 \\ 8.001875 \\ 8.02875625 \\ 5.786135938 \end{bmatrix},
$$

即

$$
\begin{bmatrix}
9 & 0 & 3.75 & 0 \\
0 & 3.75 & 0 & 2.765625 \\
3.75 & 0 & 2.765625 & 0 \\
0 & 2.765625 & 0 & 2.387695313
\end{bmatrix}
\begin{bmatrix} a_3 \\ b_3 \\ c_3 \\ d_3 \end{bmatrix} =
\begin{bmatrix} 18.5601 \\ 8.001875 \\ 8.02875625 \\ 5.786135938 \end{bmatrix}.
$$

解之得

$$
\begin{bmatrix} a_3 \\ b_3 \\ c_3 \\ d_3 \end{bmatrix} = \begin{bmatrix} 1.9599 \\ 2.4272 \\ 0.2455 \\ -0.3881 \end{bmatrix}.
$$

故得最小二乘拟合三次多项式为

$$y = 1.9599 + 2.4272x + 0.2455x^2 - 0.3881x^3.$$

12. 用最小二乘原理求一个形如 $y = a + bx^2$ 的经验公式，使其与下列数据相拟合.

x_i	19	25	31	38	44
y_i	19.0	32.3	49.0	73.3	97.8

解 由以上数据得超定方程组

$$\begin{cases} 19.0 = a + 361b, \\ 32.3 = a + 625b, \\ 49.0 = a + 961b, \\ 73.3 = a + 1444b, \\ 97.8 = a + 1936b. \end{cases}$$

其矩阵形式为 $\boldsymbol{AX} = \boldsymbol{Y}$，其中

$$\boldsymbol{A} = \begin{bmatrix} 1 & 361 \\ 1 & 625 \\ 1 & 961 \\ 1 & 1444 \\ 1 & 1936 \end{bmatrix}, \quad \boldsymbol{X} = (a, b)^{\mathrm{T}}, \quad \boldsymbol{Y} = \begin{bmatrix} 19.0 \\ 32.3 \\ 49.0 \\ 73.3 \\ 97.8 \end{bmatrix}.$$

建立法方程组

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{AX} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{Y}.$$

其中 $$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} = \begin{bmatrix} 5 & 5327 \\ 5327 & 7277699 \end{bmatrix}, \quad \boldsymbol{A}^{\mathrm{T}}\boldsymbol{Y} = \begin{bmatrix} 271.4 \\ 369321.5 \end{bmatrix},$$

即 $$\begin{bmatrix} 5 & 5327 \\ 5327 & 7277699 \end{bmatrix}\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 2714 \\ 369321.5 \end{bmatrix},$$

解之得 $$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 0.9726 \\ 0.0500 \end{bmatrix}.$$

故得经验公式为 $$y = 0.9726 + 0.05x^2.$$

13. 用最小二乘原理求一个形如 $y = ae^{bx}$ 的经验公式，使其与下表内数据相拟合.

x_i	1	2	3	4
y_i	60	30	20	10

解 因为 $$y = ae^{bx},$$

所以 $$\ln y = \ln a + bx.$$

令 $$Y = \ln y, \quad A = \ln a, \quad B = b, \quad X = x,$$

则 $$Y = A + BX.$$

由题意构造表格如下：

i	x_i	y_i	X_i	Y_i	X_i^2	X_iY_i
1	1	60	1	ln60	1	ln60
2	2	30	2	ln30	4	ln900
3	3	20	3	ln20	9	ln800 0
4	4	10	4	ln10	16	ln100 00
$\sum$			10	ln360 000	30	$\ln(4.32\times10^{12})$

由此建立法方程组

$$\begin{bmatrix}4 & 10\\10 & 30\end{bmatrix}\begin{bmatrix}A\\B\end{bmatrix}=\begin{bmatrix}\ln 360\,000\\ \ln(4.32\times10^{12})\end{bmatrix}.$$

解之得　$A=4.643\,650\,707=4.643\,7$，　$B=0.578\,074\,351=-0.578\,1$.

所以　　　$a=e^A=e^{4.643\,650\,707}=103.923$，　$b=B=-0.578\,1$.

故得经验公式为

$$y=103.923e^{-0.578\,1x}.$$

14. 求函数 $y=\sqrt{x}$ 在区间$[\frac{1}{4},1]$上的最小二乘一次多项式.

解　根据 $y=\sqrt{x}$ 得$[\frac{1}{4},1]$区间上的数表：

x_i	0.25	0.40	0.55	0.70	0.85	1
y_i	0.5	0.632 5	0.741 6	0.836 7	0.922 0	1

设一次多项式为 $y=a+bx$，再列表计算法方程组.

i	x_i	y_i	x_i^2	x_iy_i
1	0.25	0.5	0.062 5	0.125
2	0.4	0.632 5	0.16	0.253
3	0.55	0.741 6	0.302 5	0.407 88
4	0.7	0.836 7	0.49	0.585 69
5	0.85	0.922 0	0.722 5	0.783 7
6	1	1	1	1
$\sum$	3.75	4.632 8	2.737 5	3.155 3

所以
$$\begin{bmatrix}6 & 3.75\\3.75 & 2.737\,5\end{bmatrix}\begin{bmatrix}a\\b\end{bmatrix}=\begin{bmatrix}4.632\,8\\3.155\,3\end{bmatrix}.$$

解之得　$a=0.359\,752\,381=0.359\,8$，　　$b=0.659\,809\,523=0.659\,8$.

故得函数 $y=\sqrt{x}$ 在区间$[\frac{1}{4},1]$上的最小二乘一次多项式为

$$y=0.359\,8+0.659\,8x.$$

习题 7 解答

1. 分别用复化梯形公式，复化 Simpson 公式计算下列积分，并保留 9 位有效数字.

(1) $\int_0^1 \frac{x}{4+x^2}dx \quad (n=8)$.

解 设 $f(x)=\frac{x}{4+x^2}$，列表如下：

x	0	1/8	2/8	3/8	4/8
$f(x)$	0	0.031 128 404	0.061 538 461	0.090 566 037	0.117 647 058
x	5/8	6/8	7/8	1	—
$f(x)$	0.142 348 754	0.164 383 561	0.183 606 557	0.2	—

① 复化梯形公式

$$T_8=[(1/8)/2][0+0.2+2\times(0.031\,128\,404+0.061\,538\,461+0.090\,566\,037+0.117\,647\,058+0.142\,348\,754+0.164\,383\,561+0.183\,606\,557)]$$
$$=0.111\,402\,354.$$

② 复化抛物线公式

$$S_4=[(1/4)/6][0+0.2+4\times(0.031\,128\,404+0.090\,566\,037+0.142\,348\,754+0.183\,606\,557)+2\times(0.061\,538\,461+0.117\,647\,058+0.164\,383\,561)]$$
$$=0.111\,572\,382.$$

(2) $\int_0^1 \sqrt{x}\,dx \quad (n=10)$.

解 设 $f(x)=\sqrt{x}$，列表如下：

x	0	1/10	2/10	3/10	4/10	5/10
$f(x)$	0	0.316 227 766	0.447 213 595	0.547 722 557	0.632 455 532	0.707 106 781
x	6/10	7/10	8/10	9/10	1	—
$f(x)$	0.774 596 669	0.836 660 026	0.894 427 191	0.948 683 298	1	—

① 复化梯形公式

$T_{10} = [(1/10)/2][0+1+2\times(0.316\,227\,766+0.447\,213\,595+0.547\,722\,557$
$+0.632\,455\,532+0.707\,106\,781+0.774\,596\,669+0.836\,660\,026+0.894\,427\,191$
$+0.948\,683\,298)]$
$=0.660\,509\,343.$

② 复化 Simpson 公式

$S_5 = [(1/5)/6][0+1+4\times(0.316\,227\,766+0.547\,722\,557+0.707\,106\,781+0.836\,660\,026$
$+0.948\,683\,298)+2\times(0.447\,213\,595+0.632\,455\,532+0.774\,596\,669+0.894\,427\,191)]$
$=0.664\,099\,575.$

(3) $\int_0^1 e^{-x^2}dx \quad (n=10).$

解 设 $f(x)=e^{-x^2}$，列表如下：

x	0	1/10	2/10	3/10	4/10	5/10
$f(x)$	1	0.990 049 833	0.960 789 439	0.913 931 185	0.852 143 789	0.778 800 783
x	6/10	7/10	8/10	9/10	1	—
$f(x)$	0.697 676 326	0.612 626 394	0.527 292 424	0.444 858 066	0.367 879 441	—

① 复化梯形公式

$T_{10} = [(1/10)/2][1+0.367\,879\,441+2\times(0.990\,049\,833+0.960\,789\,439$
$+0.913\,931\,185+0.852\,143\,789+0.778\,800\,783+0.697\,676\,326+0.612\,626\,394$
$+0.527\,292\,424+0.444\,858\,066)]$
$=0.746\,210\,796.$

② 复化 Simpson 公式

$S_5 = [(1/5)/6][1+0.367\,879\,441+4\times(0.990\,049\,833+0.913\,931\,185+0.778\,800\,783$
$+0.612\,626\,394+0.444\,858\,066)+2\times(0.960\,789\,439+0.852\,143\,789+0.697\,676\,326$
$+0.527\,292\,424]$
$=0.746\,742\,37.$

(4) $\int_0^{\frac{\pi}{2}} \sqrt{4-\sin^2 x}\,dx \quad (n=6).$

解 设 $f(x)=\sqrt{4-\sin^2 x}$，列表如下：

x	0	$\pi/36$	$2\pi/36$	$3\pi/36$	$4\pi/36$	$5\pi/36$	$\pi/6$
$f(x)$	2	1.998 100 067	1.992 447 317	1.983 182 468	1.970 538 561	1.954 838 562	1.936 491 673

① 复化梯形公式

$T_6=[(\pi/36)/2][2+1.936\,491\,673+2\times(1.998\,100\,067+1.992\,447\,317+1.983\,182\,468+1.970\,538\,561+1.954\,838\,562)]$

$=1.035\,621\,9.$

② 复化 Simpson 公式

$S_3=[(\pi/18)/6][2+1.936\,491\,673+4\times(1.998\,100\,067+1.983\,182\,468+1.954\,838\,562)+2\times(1.992\,447\,317+1.970\,538\,561)]$

$=1.035\,764\,127$

(5) $\int_0^{\frac{\pi}{2}}\frac{\sin x}{x}dx\quad(n=8).$

解 设 $f(x)=\frac{\sin x}{x}$，列出下列数表：

x	0	$\pi/16$	$2\pi/16$	$3\pi/16$	$4\pi/16$
$f(x)$	1	0.993 586 851	0.974 495 358	0.943 165 32	0.900 316 316
x	$5\pi/16$	$6\pi/16$	$7\pi/16$	$\pi/8$	—
$f(x)$	0.846 927 992	0.784 213 303	0.713 585 487	0.636 619 772	—

① 复化梯形公式

$T_8=[(\pi/16)/2][1+0.636\,619\,772+2\times(0.993\,586\,851+0.974\,495\,358+0.943\,165\,32+0.900\,316\,316+0.846\,927\,992+0.784\,213\,303+0.713\,585\,487)]$

$=1.369\,459\,608.$

② 复化 Simpson 公式

$S_4=[(\pi/8)/6][1+0.636\,619\,772+4\times(0.993\,586\,851+0.943\,165\,32+0.846\,927\,992+0.713\,585\,487)+2\times(0.974\,495\,358+0.900\,316\,316+0.784\,213\,303)]$

$=1.370\,764\,075.$

2. 试用 Romberg 求积公式计算积分 $\frac{2}{\sqrt{\pi}}\int_0^1 e^{-x^2}dx$. 要求误差不超过 0.5×10^{-5}.

解 先计算积分 $\int_0^1 e^{-x^2}dx$ 的值.

令 $f(x)=e^{-x^2}$，计算：

$$T_0^{(0)} = \frac{1-0}{2}[f(0)+f(1)] = \frac{1}{2}[e^0+e^{-1}] = 0.683\,939\,72,$$

$$T_0^{(1)} = \frac{1}{2}T_0^{(0)} + \frac{1-0}{2}f(\frac{1+0}{2}) = \frac{1}{2}\times 0.683\,939\,72 + \frac{1}{2}e^{-(1/4)} = 0.731\,370\,251,$$

$$T_1^{(0)} = \frac{4}{3}T_0^{(1)} - \frac{1}{3}T_0^{(0)} = 0.747\,180\,428,$$

$$T_0^{(2)} = \frac{1}{2}T_0^{(1)} + \frac{1}{2^2}(e^{-(1/4)^2} + e^{-(3/4)^2}) = 0.742\,984\,097,$$

$$T_1^{(1)} = \frac{4}{3}T_0^{(2)} - \frac{1}{3}T_0^{(1)} = 0.746\,855\,379,$$

$$T_2^{(0)} = \frac{16}{15}T_1^{(1)} - \frac{1}{15}T_1^{(0)} = 0.746\,833\,709,$$

$$T_0^{(3)} = \frac{1}{2}T_0^{(2)} + \frac{1}{2^3}[e^{-(1/8)^2} + e^{-(3/8)^2} + e^{-(7/8)^2}] = 0.745\,865\,614,$$

$$T_1^{(2)} = \frac{4}{3}T_0^{(3)} - \frac{1}{3}T_0^{(2)} = 0.746\,826\,12,$$

$$T_2^{(1)} = \frac{16}{15}T_1^{(2)} - \frac{1}{15}T_1^{(1)} = 0.746\,824\,169,$$

$$T_3^{(0)} = \frac{64}{63}T_2^{(1)} - \frac{1}{63}T_2^{(0)} = 0.746\,824\,018,$$

$$T_0^{(4)} = \frac{1}{2}T_0^{(3)} + \frac{1}{2^4}[e^{-(1/16)^2} + e^{-(3/16)^2} + e^{-(5/16)^2} + e^{-(7/16)^2} + e^{-(9/16)^2} + e^{-(11/16)^2} + e^{-(13/16)^2} + e^{-(15/16)^2}],$$
$$= 0.746\,584\,596,$$

$$T_1^{(3)} = \frac{4}{3}T_0^{(4)} - \frac{1}{3}T_0^{(3)} = 0.746\,824\,257,$$

$$T_2^{(2)} = \frac{16}{15}T_1^{(3)} - \frac{1}{15}T_1^{(2)} = 0.746\,824\,132,$$

$$T_3^{(1)} = \frac{64}{63}T_2^{(2)} - \frac{1}{63}T_2^{(1)} = 0.746\,824\,132,$$

$$T_4^{(0)} = \frac{256}{255}T_3^{(1)} - \frac{1}{255}T_3^{(0)} = 0.746\,824\,132.$$

将上述计算值列表：

二分次数 i	等份数 2^i	$T_0^{(i)}$	$T_1^{(i)}$	$T_2^{(i)}$	$T_3^{(i)}$	$T_4^{(i)}$
0	1	0.683 939 72				
1	2	0.731 370 251	0.747 180 428			
2	4	0.742 984 09	0.746 855 379	0.746 833 709		
3	8	0.745 865 614	0.746 826 12	0.746 824 169	0.746 824 018	
4	16	0.746 584 596	0.746 824 25	0.746 824 132	0.746 824 132	0.746 824 132

因为 $$\frac{2}{\sqrt{\pi}}(T_4^{(0)}-T_3^{(0)})=0.000\,000\,128<0.5\times10^{-5},$$

所以 $$\frac{2}{\sqrt{\pi}}\int_0^1 e^{-x^2}dx=\frac{2}{\sqrt{\pi}}\times T_4^{(0)}=0.842\,700\,792\approx0.842\,70.$$

3. 试分别用梯形公式、Simpson 公式与 Romberg 算法计算积分 $\int_0^1 0.5\sqrt{x}\,dx$.

解 ① 梯形公式：

$$T=\frac{1-0}{2}[f(0)+f(1)]=\frac{1}{2}[0.5\sqrt{0}+0.5\sqrt{1}]=0.25;$$

②Simpson 公式：

$$S=\frac{1-0}{6}(0.5\sqrt{0}+0.5\sqrt{1}+4\times0.5\sqrt{0.5})=0.319\,035\,593;$$

③Romberg 算法：

$$T_1=\frac{1-0}{2}(0.5\sqrt{0}+0.5\sqrt{1})=0.25,$$

$$T_2=\frac{1}{2}T_1+\frac{1}{2}f(0.5)=\frac{1}{2}\times0.25+\frac{1}{2}\times0.5\sqrt{0.5}=0.301\,776\,695,$$

$$T_4=\frac{1}{2}T_2+\frac{1}{2^2}(0.5\sqrt{1/4}+0.5\sqrt{3/4})=0.321\,641\,523,$$

$$T_8=\frac{1}{2}T_4+\frac{1}{2^3}(0.5\sqrt{1/8}+0.5\sqrt{3/8}+0.5\sqrt{5/8}+0.5\sqrt{7/8})=0.329\,065\,11,$$

$$S_1=\frac{4}{3}T_2-\frac{1}{3}T_1=0.319\,035\,593,$$

$$S_2=\frac{4}{3}T_4-\frac{1}{3}T_2=0.328\,263\,132,$$

$$S_4=\frac{4}{3}T_8-\frac{1}{3}T_4=0.331\,539\,64,$$

$$C_1=\frac{16}{15}S_2-\frac{1}{15}S_1=0.328\,878\,301,$$

$$C_2=\frac{16}{15}S_4-\frac{1}{15}S_2=0.331\,758\,073,$$

$$R_1=\frac{64}{63}C_2-\frac{1}{63}C_1=0.331\,803\,784.$$

4. 试确定系数 A_1,A_2,A_3，使求积公式

$$\int_{-1}^1 f(x)dx\approx A_1f(-1)+A_2f(-\frac{1}{3})+A_3f(\frac{1}{3})$$

具有 2 次代数精度.

解 令 $f(x)=1$，则

$$2=A_1+A_2+A_3, \qquad ①$$

令 $f(x)=x$，则

$$0=A_1-(1/3)\times A_2+(1/3)\times A_3, \qquad ②$$

令 $f(x)=x^2$,则

$$2/3 = A_1 + (1/9)\times A_2 + (1/9)\times A_3. \qquad ③$$

联立式①、②、③得

$$A_1 = 1/2,\quad A_2 = 0,\quad A_3 = 3/2,$$

所以
$$\int_{-1}^{1} f(x)\mathrm{d}x \approx \frac{1}{2}f(-1) + \frac{3}{2}f\left(\frac{1}{3}\right).$$

再令 $f(x)=x^3$ 代入上式得

$$0 \neq -\frac{1}{2} + \frac{1}{18}.$$

所以,所求数值求积公式具有二次代数精度.

5. 利用 Gauss-Legendre 求积公式计算积分$\int_{-4}^{4}\frac{1}{1+x^2}\mathrm{d}x$. 分别取 $n=1,2,3$,并与真实值 $2\arctan 4 \approx 2.651\,635\,4$ 比较.

解 ① 取 $n=1$,则

$$\begin{aligned}\int_{-4}^{4}\frac{1}{1+x^2}\mathrm{d}x &= \int_{-1}^{1}\frac{4}{1+16t^2}\mathrm{d}t \\ &= \frac{4}{1+16\times(-1/\sqrt{3})^2} + \frac{4}{1+16\times(1/\sqrt{3})^2} \\ &= 1.263\,157\,895;\end{aligned}$$

② 取 $n=2$,则

$$\begin{aligned}\int_{-4}^{4}\frac{1}{1+x^2}\mathrm{d}x &= \int_{-1}^{1}\frac{4}{1+16t^2}\mathrm{d}t \\ &= 2\times 0.555\,555\,556\times 4/(1+16\times 0.774\,596\,669^2) \\ &\quad + 0.888\,888\,889\times 4/(1+16\times 0^2) \\ &= 3.974\,842\,768;\end{aligned}$$

③ 取 $n=3$,则

$$\begin{aligned}\int_{-4}^{4}\frac{1}{1+x^2}\mathrm{d}x &= \int_{-1}^{1}\frac{4}{1+16t^2}\mathrm{d}t \\ &= 2\times 0.347\,854\,845\times 4/(1+16\times 0.861\,136\,312^2) \\ &\quad + 2\times 0.652\,145\,155\times 4/(1+16\times 0.339\,981\,044^2) \\ &= 2.047\,285\,006.\end{aligned}$$

习题 8 解答

1. 已知初值问题

$$\begin{cases} y' = \dfrac{1}{1+x^2} - 2y^2, & 0 \leqslant x \leqslant 1, \\ y(0) = 0. \end{cases}$$

试用 Euler 法，取不同步长 0.1 和 0.2，分别计算到 $y(1)$，并与准确解

$$y(x) = \frac{x}{1+x^2}$$

相比较. 近似值和准确值都取四位有效数字.

解 (1) 取 $h = 0.1$，Euler 公式为

$$\begin{cases} y_{n+1} = y_n + hK, \\ K = f(x_n, y_n) = \dfrac{1}{1+x_n^2} - 2y_n^2. \end{cases}$$

计算结果列表如下：

x_n	y_n	K	$y(x_n)$
0.000 0	0.000 0	1.000 0	0.000 0
0.100 0	0.100 0	0.970 1	0.099 0
0.200 0	0.197 0	0.883 9	0.192 3
0.300 0	0.285 4	0.754 5	0.275 2
0.400 0	0.360 9	0.601 6	0.344 8
0.500 0	0.421 1	0.445 3	0.400 0
0.600 0	0.465 6	0.301 7	0.441 2
0.700 0	0.495 8	0.179 5	0.469 8
0.800 0	0.513 8	0.081 8	0.487 8
0.900 0	0.522 0	0.007 5	0.497 2
1.000 0	0.522 8	—	0.500 0

(2) 取 $h = 0.2$，Euler 公式为

$$\begin{cases} y_{n+1} = y_n + hK, \\ K = f(x_n, y_n) = \dfrac{1}{1+x_n^2} - 2y_n^2. \end{cases}$$

计算结果列表如下：

x_n	y_n	K	$y(x_n)$
0.000 0	0.000 0	1.000 0	0.000 0
0.200 0	0.200 0	0.881 5	0.192 3
0.400 0	0.376 3	0.578 9	0.344 8
0.600 0	0.492 1	0.251 0	0.441 2
0.800 0	0.542 3	0.021 6	0.487 8
1.000 0	0.544 6	—	0.500 0

2. 用改进的 Euler 公式和四阶经典 R－K 法，取步长 $h=0.1$，计算下列初值问题的近似解，并与精确值比较，且二者均保留四位小数.

$$\begin{cases} y'=x+y, & 0\leqslant x\leqslant 1, \\ y(0)=1. \end{cases}$$

此问题的精确解为 $y(x)=-x-1+2e^{x}$.

解 (1) 改进的 Euler 法($h=0.1$)为

$$\begin{cases} y_{n+1}=y_n+hK^{*}, \\ K^{*}=\dfrac{1}{2}(K_1+K_2), \\ K_1=f(x_n,y_n)=x_n+y_n, \\ K_2=f(x_n+h,y_n+hK_1)=(x_n+h)+(y_n+hK_1). \end{cases}$$

计算结果列表如下：

x_n	y_n	$K_i(i=1,2)$	K^{*}	$y(x_n)$
0.000 0	1.000 0	1.000 0		1.000 0
0.100 0	1.100 0	1.200 0	1.100 0	
0.100 0	1.110 0	1.210 0		1.110 3
0.200 0	1.231 0	1.431 0	1.320 5	
0.200 0	1.242 1	1.442 1		1.242 8
0.300 0	1.386 3	1.686 3	1.564 2	
0.300 0	1.398 5	1.698 5		1.399 7
0.400 0	1.168 4	1.968 4	1.833 5	
0.400 0	1.581 9	1.981 9		1.583 6
0.500 0	1.780 1	2.280 1	2.131 0	
0.500 0	1.795 0	2.295 0		1.797 4
0.600 0	2.024 5	2.624 5	2.459 8	
0.600 0	2.041 0	2.641 0		2.044 2
0.700 0	2.305 1	3.005 1	2.823 1	
0.700 0	2.323 3	3.023 3		2.327 5

续表

x_n	y_n	$K_i(i=1,2)$	K^*	$y(x_n)$
0.800 0	2.625 6	3.425 6	3.224 5	
0.800 0	2.645 8	3.445 8		2.651 1
0.900 0	2.990 4	3.890 4	3.668 1	
0.900 0	3.012 6	3.912 6		3.019 2
1.000 0	3.403 9	4.403 9	4.158 3	
1.000 0	3.428 4			3.436 6

(2) 四阶经典 R－K 法($h=0.1$) 为

$$\begin{cases} y_{n+1}=y_n+hK^*, \\ K^*=\frac{1}{6}(K_1+2K_2+2K_3+K_4), \\ K_1=f(x_n,y_n)=x_n+y_n, \\ K_2=f(x_n+\frac{1}{2}h,y_n+\frac{1}{2}hK_1)=(x_n+\frac{1}{2}h)+(y_n+\frac{1}{2}hK_1), \\ K_3=f(x_n+\frac{1}{2}h,y_n+\frac{1}{2}hK_2)=(x_n+\frac{1}{2}h)+(y_n+\frac{1}{2}hK_2), \\ K_4=f(x_n+h,y_n+hK_3)=(x_n+h)+(y_n+hK_3). \end{cases}$$

计算结果列表如下：

x_n	y_n	$k_i(i=1,2,3,4)$	K^*	$y(x_n)$
0.000 0	1.000 0	1.000 0		1.000 0
0.050 0	1.050 0	1.100 0		
0.050 0	1.055 0	1.105 0		
0.100 0	1.110 5	1.210 5	1.103 4	
0.100 0	1.110 3	1.210 3		1.110 3
0.150 0	1.170 8	1.320 8		
0.150 0	1.176 3	1.326 3		
0.200 0	1.242 9	1.442 9	1.324 6	
0.200 0	1.242 8	1.442 8		1.242 8
0.250 0	1.314 9	1.546 9		
0.250 0	1.321 0	1.571 0		
0.300 0	1.399 9	1.699 9	1.569 1	
0.300 0	1.399 7	0.699 7		1.399 7
0.350 0	1.484 7	1.834 7		
0.350 0	1.491 4	1.841 4		
0.400 0	1.583 8	1.983 8	1.839 3	
0.400 0	1.583 6	1.983 6		1.583 6

续表

x_n	y_n	$K_i(i=1,2,3,4)$	K^*	$y(x_n)$
0.450 0	1.682 8	2.132 8		
0.450 0	1.690 2	2.140 2		
0.500 0	1.797 6	2.297 6	2.137 9	
0.500 0	1.797 4	2.297 4		1.797 4
0.550 0	1.192 3	2.462 3		
0.550 0	1.920 5	2.470 5		
0.600 0	2.044 5	2.644 5	2.467 9	
0.600 0	2.044 2	2.644 2		2.044 2
0.650 0	2.176 4	2.826 4		
0.650 0	2.185 5	2.835 5		
0.700 0	2.327 8	3.027 8	2.832 6	
0.700 0	2.327 5	3.027 5		2.327 5
0.750 0	2.478 9	3.228 9		
0.750 0	2.488 9	3.238 9		
0.800 0	2.651 4	3.451 4	3.235 7	
0.800 0	2.651 1	3.451 1		2.651 1
0.850 0	2.823 7	3.673 7		
0.850 0	2.834 8	3.684 8		
0.900 0	3.019 6	3.919 6	3.681 3	
0.900 0	3.019 2	3.919 2		3.019 2
0.950 0	3.215 2	4.165 2		
0.950 0	3.227 5	4.177 5		
1.000 0	3.436 9	4.436 9	4.173 6	
1.000 0	3.436 6			3.436 6

参考书目

[1] 朱长青. 数值计算方法及其应用. 北京:科学出版社,2006 年.

[2]《现代应用数学手册》编委会. 现代应用数学手册,计算:与数值分析卷[ ]. 北京:高等教育出版社.

[3] Curtis F. Gearld,Patrick O. Wheatly 原著,白峰杉改编. Applied Numerical Analysis,应用数值分析(第 7 版). 北京:高等教育出版社,2006 年.

[4] 李庆扬等,现代数值分析. 北京:高等教育出版社,1995 年.

[5] 南京大学数学计算数学专业,数值逼近方法. 北京:科学出版社,1978 年.

[6] 陈德创. 多元数据处理. 北京:化学工业出版社,1998 年.

[7] 常兆光等,随机数据处理方法. 北京:石油大学出版社,1997 年.